

# **UNIVERSIDAD POLITÉCNICA SALESIANA**

**FACULTAD DE INGENIERIAS**

**SEDE QUITO-CAMPUS SUR**

**CARRERA DE INGENIERIA DE SISTEMAS**

**MENCIÓN INFORMÁTICA PARA LA GESTIÓN**

**DISEÑO, DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA  
DE CONTROL E INVENTARIO PARA EL CENTRO DE DIFUSIÓN Y  
DOCUMENTACIÓN DEL CONSEJO NACIONAL DE LAS MUJERES**

**TESIS PREVIA A LA OBTENCIÓN DEL TÍTULO DE INGENIERO DE SISTEMAS**

**ANA LUCÍA GUANOCHANGA GUZMÁN  
DIEGO FERNANDO RUEDA MACAS**

**DIRECTOR ING. RENÉ ARÉVALO**

**Quito, julio 2011**

## **DECLARACIÓN**

Nosotros, Ana Lucía Guanochanga Guzmán y Diego Fernando Rueda Macas, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y que hemos consultado las referencias bibliográficas que incluyen en este documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondiente a este trabajo a la Universidad Politécnica Salesiana, según lo establecido por la Ley de Propiedad Intelectual, por su reglamento y por la normatividad institucional vigente.

Ana Lucía  
Guanochanga Guzmán

Diego Fernando  
Rueda Macas

## **CERTIFICACIÓN**

Certifico que el presente trabajo fue desarrollado por Ana Lucía Guanochanga Guzmán y Diego Fernando Rueda Macas, bajo mi dirección.

---

Ing. René Arévalo  
Director de Tesis

## **AGRADECIMIENTOS**

En primer lugar quisiéramos agradecer a nuestro tutor de tesis, Ing. René Arévalo, por su incondicional guía a través de todo este proceso; sin su energía, conocimiento, paciencia y apoyo el desarrollo de nuestro proyecto no habría tenido éxito.

Al personal del Departamento de Informática del Consejo Nacional de las Mujeres, pero principalmente a la Servidora Pública 4, la Ing. Mayra Torres, por todos sus invaluable comentarios y apoyo en el desarrollo de nuestro sistema.

Merecen también de toda nuestra gratitud aquellas personas que nos apoyaron e impulsaron para completar nuestro proyecto: Gabriela Matapuncho y Fernando Pérez.

Finalmente, lo más importante: agradecer profundamente a nuestras familias por su eterno apoyo, comprensión y amor.

## CONTENIDO

	<b>pág.</b>
LISTA DE ILUSTRACIONES	X
LISTA DE TABLAS	XIII
LISTA DE ANEXOS	XV
 <b>CAPÍTULO 1: INTRODUCCION</b>	 1
1.1. Descripción del problema	2
1.2. Objetivo general	2
1.3. Objetivos específicos	2
1.4. Alcance y limitaciones	3
1.5. Justificación	7
1.6. Situación actual	8
1.7. Solución propuesta	8
1.8. Metodología	9
1.9. Hardware y software a utilizar	13
 <b>CAPÍTULO 2: MARCO TEÓRICO</b>	 14
2.1. Introducción a Php	14
2.1.1. Variables	16
2.1.2. Operadores	19
2.1.3. Estructuras de Control	26
2.1.4. Funciones	31
2.2. Aplicaciones de seguridad en Php	35
2.2.1. Sesiones en Php	35
2.3. Introducción a JavaScript	37
2.3.1. Breve historia	38

	VII
2.3.2. Sintaxis	39
2.3.3. JavaScript y otros navegadores	40
2.3.4. JavaScript en otros entornos	41
2.3.5. Variables	41
2.3.6. Tipos de variables	42
2.3.7. Operadores	44
2.3.8. Estructuras de control de flujo	48
2.3.9. Funciones y propiedades básicas de JavaScript	51
2.4. Lenguaje de modelamiento unificado-UML	57
2.4.1. Jerarquía de los diagramas UML	57
<b>CAPÍTULO 3: ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE</b>	60
3.1. Requerimientos Funcionales	60
3.2. Requerimientos no Funcionales	63
<b>CAPÍTULO 4: ANÁLISIS Y DISEÑO DEL SISTEMA</b>	66
4.1. Especificaciones de Caso de Uso	66
4.1.1. Características de los usuarios	66
4.1.2. Roles y tareas de los usuarios	67
4.1.3. Especificación de escenarios	68
4.2. Diseño del Sistema	85
4.2.1. Archivos o Base de Datos	85
<b>CAPÍTULO 5: BASE DE DATOS POSTGRESQL</b>	87
5.1. Introducción al PostgreSQL	87
5.1.1. Breve historia	87
5.1.2. Terminología	88
5.2. Constantes	89

	VIII
5.2.1. Constantes tipo cadena	89
5.2.2. Constantes tipo entero	89
5.2.3. Constantes tipo punto flotante	90
5.2.4. Constantes de tipo definido por el usuario	90
5.2.5. Constantes de tipo array	90
5.3. Secuencias	91
5.4. Operadores	92
5.5. Parámetros	93
5.5.1. Expresiones Funcionales	93
5.5.2. Expresiones de Agregación	93
5.5.3. Lista Objetivo	94
5.5.4. Calificadores	94
5.6. Instalación del PostgreSQL	94
5.7. Configuración del PostgreSQL	102
<b>CAPÍTULO 6: DIAGRAMAS</b>	105
6.1. Diagramas de Interacción	105
6.1.1. Diagramas de Secuencia	105
6.2. Diagrama de Comportamiento	110
6.2.1. Diagrama de Casos de Uso del Negocio	111
6.2.2. Diagrama de Actividades	113
6.3. Diagramas Estructurales	116
6.3.1. Diagrama de Componentes	116
<b>CAPÍTULO 7: PRUEBAS Y CONTROL DE CALIDAD</b>	118
7.1. Pruebas	118
7.2. Control de calidad	121
<b>CAPÍTULO 8: CONCLUSIONES Y RECOMENDACIONES</b>	131

**REFERENCIAS BIBLIOGRÁFICAS**

125

**ANEXOS**

128



## LISTA DE ILUSTRACIONES

	<b>pág.</b>
<b>Figura 1.</b> Esquema de solicitud de página al servidor	16
<b>Figura 2.</b> Ingresar al sistema	69
<b>Figura 3.</b> Ingresar Funcionaria	70
<b>Figura 4.</b> Crear Tipo de transacción	71
<b>Figura 5.</b> Crear Usuario Final	72
<b>Figura 6.</b> Crear Tipo de Documento	73
<b>Figura 7.</b> Ingresar Producto	74
<b>Figura 8.</b> Crear Tipo de Unidad	75
<b>Figura 9.</b> Crear Tipo de Producto	75
<b>Figura 10.</b> Ingresar Proveedor	76
<b>Figura 11.</b> Ingresar Destino Geográfico	77
<b>Figura 12.</b> Ingresar Responsable	78
<b>Figura 13.</b> Crear Tipo de Usuario	79
<b>Figura 14.</b> Crear Tipo de Servicio	80
<b>Figura 15.</b> Ingresar Organización	81
<b>Figura 16.</b> Responder Encuesta Virtual	82
<b>Figura 17.</b> Realizar Ingresos	83
<b>Figura 18.</b> Realizar Egresos	84
<b>Figura 19. Diseño Lógico</b>	85
<b>Figura 20. Diseño Físico</b>	86
<b>Figura 21.</b> Setup - PostgreSQL	95
<b>Figura 22.</b> Installation Directory	95
<b>Figura 23.</b> Data Directory	96
<b>Figura 24.</b> Password	96
<b>Figura 25.</b> Port	97
<b>Figura 26.</b> Advanced Optiont	97
<b>Figura 27.</b> Ready to install	98
<b>Figura 28.</b> Installing	98

<b>Figura 29.</b> Install	99
<b>Figura 30.</b> Completing	99
<b>Figura 31.</b> Remote Server	100
<b>Figura 32.</b> Select the applications	100
<b>Figura 33.</b> Start the instalation	101
<b>Figura 34.</b> JDBC Drivers	101
<b>Figura 35.</b> Instalation complete	102
<b>Figura 36.</b> Ingresar al sistema	105
<b>Figura 37.</b> Crear Tipo de usuario	106
<b>Figura 38.</b> Crear Tipo de Transacción	106
<b>Figura 39.</b> Ingresar Usuario Final	106
<b>Figura 40.</b> Ingresar Tipo de documento	107
<b>Figura 41.</b> Ingresar Productos	107
<b>Figura 42.</b> Ingresar Tipo de unidad	107
<b>Figura 43.</b> Ingresar Tipo de producto	108
<b>Figura 44.</b> Ingresar Proveedor	108
<b>Figura 45.</b> Ingresar Destino geográfico	108
<b>Figura 46.</b> Ingresar Responsable	109
<b>Figura 47.</b> Ingresar Organización	109
<b>Figura 48.</b> Crear Tipo de servicio	109
<b>Figura 49.</b> Ingresar Funcionaria	110
<b>Figura 50.</b> Responder Encuesta Virtual	110
<b>Figura 51.</b> Verificar existencias en bodega	111
<b>Figura 52.</b> Recepción de mercadería del proveedor	111
<b>Figura 53.</b> Recepción de material de Organizaciones:	112
<b>Figura 54.</b> Almacenar el material	112
<b>Figura 55.</b> Entregar material	113
<b>Figura 56.</b> Verificar existencias en bodega	114
<b>Figura 57.</b> Recepción de mercadería de proveedor	114
<b>Figura 58.</b> Recepción de material de Organizaciones	115
<b>Figura 59.</b> Almacenar el material	115

<b>Figura 60.</b> Entregar los materiales	116
<b>Figura 61.</b> Diagrama de Componentes	117
<b>Figura 62. Funcionaria</b>	126
<b>Figura 63. Productos</b>	127
<b>Figura 64. Cabecera</b>	128

## LISTA DE TABLAS

	<b>pág.</b>
<b>Tabla 1.</b> Precedencia de Operadores	19
<b>Tabla 2.</b> Operadores Aritméticos	20
<b>Tabla 3.</b> Operadores de Asignación	21
<b>Tabla 4.</b> Operadores bit a bit	21
<b>Tabla 5.</b> Operadores de Comparación	22
<b>Tabla 6.</b> Operadores de Incremento / Decremento	24
<b>Tabla 7.</b> Operadores Lógicos	25
<b>Tabla 8.</b> Tabla de Conversión	43
<b>Tabla 9.</b> Negación	46
<b>Tabla 10.</b> AND	46
<b>Tabla 11.</b> OR	47
<b>Tabla 12.</b> Ingresar al sistema	70
<b>Tabla 13.</b> Ingresar Funcionaria	71
<b>Tabla 14.</b> Crear Tipos de Transacción	72
<b>Tabla 15.</b> Crear Usuario Final	73
<b>Tabla 16.</b> Crear Tipo de Documento	73
<b>Tabla 17.</b> Ingresar Producto	74
<b>Tabla 18.</b> Crear Tipo de Unidad	75
<b>Tabla 19.</b> Crear Tipo de Producto	76
<b>Tabla 20.</b> Ingresar Proveedor	77
<b>Tabla 21.</b> Ingresar Destino Geográfico	78
<b>Tabla 22.</b> Ingresar Responsable	79
<b>Tabla 23.</b> Crear Tipo de Usuario	79
<b>Tabla 24.</b> Crear Tipo de Servicio	80
<b>Tabla 25.</b> Ingresar Organización	81
<b>Tabla 26.</b> Responder Encuesta Virtual	82
<b>Tabla 27.</b> Realizar Ingresos	83

	XIV
<b>Tabla 28. Realizar Egresos</b>	84
<b>Tabla 29. Verificar la implementación del acceso al sistema</b>	119
<b>Tabla 30. Verificar la confirmación de cierre de sesión</b>	119
<b>Tabla 31. Verificar la implementación del menú opciones</b>	120
<b>Tabla 32. Verificar la lectura con la base de datos</b>	120
<b>Tabla 33. Factor de calidad</b>	124
<b>Tabla 34. Funcionaria</b>	126
<b>Tabla 35. Productos</b>	127
<b>Tabla 36. Egresos</b>	128
<b>Tabla 37. Movimientos por destinatario</b>	129
<b>Tabla 38. Egresos Totales</b>	130
<b>Tabla 39. Funcionaria</b>	130
<b>Tabla 40. Producto</b>	130

## LISTA DE ANEXOS

**Anexo A** Manual de Usuario

**Anexo B** Manual Técnico

**Anexo C** Manual de instalación y configuración de PostgreSQL

## RESUMEN

El proyecto con tema **“Diseño, Desarrollo e Implementación de un Sistema de Control e Inventario para el Centro de Difusión y Documentación del Consejo Nacional de las Mujeres”** busca establecer la problemática así como su solución en el ingreso y egreso de los datos sobre los materiales donados y difundidos que producen el traspapeleo e ingreso erróneo de documentos a la organización, de esta forma optimizando de mejor manera en el manejo del inventario sobre los materiales con los que se cuenta y se trabaja en el Centro de Difusión y Documentación, mediante el uso de formularios de fácil empleo para la funcionaria el cuál agilizará la administración del control e inventario de los materiales con los que se trabaja. También establecer solución sobre la falta de reportes estadísticos acordes a las necesidades de la institución, presentados de una manera entendible tanto las consultas o datos desplegados como su respectivo gráfico. El proyecto de tesis tiene como objetivo brindar una solución a corto plazo así como mejorar, agilizar y facilitar los procesos de control e inventario para el Centro de Difusión y Donación del Consejo Nacional de las Mujeres, haciendo uso de las herramientas de software libre.

# CAPÍTULO 1

## 1. INTRODUCCIÓN

La Facultad de Ingenierías de la Universidad Politécnica Salesiana con el fin de incentivar a su alumnado en el desarrollo de soluciones informáticas, acordes a las problemáticas y necesidades actuales de las diferentes instituciones del país, promueve el desarrollo de proyectos para aportar con soluciones a las instituciones beneficiándose de esta forma el estudiante y la institución para la que se lleva a efecto el desarrollo de la solución.

Actualmente las instituciones del estado en busca de agilizar los mantenimientos a sus respectivos sistemas y el ahorro en la compra de licencias se han sujetado a una modernización en sus procesos informáticos, reemplazándolos por el uso de herramientas de software libre en sus diversos sistemas de aplicación y gestión. De acuerdo a las necesidades que se han suscitado dentro del Consejo Nacional de las Mujeres se presenta el siguiente tema “**Diseño, Desarrollo e Implementación de un Sistema de Control e Inventario para el Centro de Difusión y Documentación del Consejo Nacional de las Mujeres**” el que permitirá mejorar y facilitar los procesos de esta institución, usando herramientas de software libre, dando así una solución a la problemática planteada.

Para lograr establecer las causas a la problemática en el registro de los datos sobre los materiales donados y difundidos que producen el traspapeleo e ingreso erróneo de documentos a la organización, se verificó las interfaces de ingreso de datos y los reportes con los que cuenta el sistema actual; como resultado de este análisis del sistema se establece los siguientes problemas:

- 1) Falta de reportes estadísticos de los documentos ingresados.
- 2) Carencia de registro en el sistema de datos que posibiliten la emisión de reportes de acuerdo a las necesidades de la institución.
- 3) Implementar una interface más amigable para promover y facilitar el uso del sistema por parte de las encargadas de registrar la información del sistema.



El presente desarrollo tiene por objetivo brindar una solución a corto plazo para el Centro de Difusión y Donación del Consejo Nacional de las Mujeres.

### **1.1. DESCRIPCIÓN DEL PROBLEMA**

Debido a la gran cantidad de datos que se procesan en el Centro de Difusión y Documentación del Consejo Nacional de las Mujeres y la necesidad de realizar proyecciones y las respectivas planificaciones sobre la información de material donado a instituciones del estado, embajadas, ONG's, medios de comunicación, público en general, instituciones educativas, etc., se ha visto la necesidad de realizar un sistema que apoye a las funcionarias de la institución encargadas del área de difusión y documentación, en el manejo de información de material de donación y material bibliográfico además en el análisis de datos estadísticos en una forma gráfica y entendible para realizar sus labores en una forma más ágil; reduciendo así el tiempo de elaboración de tablas, muestreos y proyecciones sobre los datos manejados.

### **1.2. OBJETIVO GENERAL**

- Desarrollar una herramienta de software que permita disponer de un adecuado control del inventario del material de difusión y donación, incluyendo los respectivos movimientos que lo afectan, usando para este fin reportes periódicos con el cual se puedan realizar proyecciones y una adecuada planificación sobre el material de difusión y donación que debe ser adquirido en el Centro de Difusión y Documentación del Consejo Nacional de las Mujeres; esta herramienta se implementará utilizando software libre.

### **1.3. OBJETIVOS ESPECÍFICOS**

- Reducir los tiempos de trabajo de las funcionarias al momento de elaborar los reportes sobre datos reales en relación a la cantidad de material difundido o consultado.

- Proveer de la facilidad para exportar los reportes y gráficos estadísticos a varios formatos de archivos; los formatos previstos son Word, Excel, archivo de imagen (bmp, gif).
- Registrar los egresos e ingresos del material de difusión para las diversas funciones a trabajar por el sistema, haciendo que la programación sea a más de organizada y estructurada.
- Diseñar interfaces amigables al usuario, que faciliten la navegación por pantallas.
- Proveer reportes de consolidados generales y detallados por materiales o productos manejados, por localidades, por Organización de Mujer a beneficiar, por Especialista, y por cualquier dato que el usuario disponga.
- Aplicar conocimientos adquiridos sobre java y php para el desarrollo de procesos y funciones propias del sistema, haciendo seguro el envío de datos al servidor y recepción de los mismos al usuario.
- Adquirir conocimientos necesarios sobre PostgreSQL para el diseño de una base de datos que trabaje de forma eficaz de acuerdo a los requerimientos y necesidades del sistema.
- Entregar un manual de usuario el cual permitirá usar de una manera fácil el sistema.
- Entregar un manual técnico (mantenimiento) y diccionario de datos que permitirá a las personas encargadas de sistemas realizar un mantenimiento, actualización y adecuaciones necesarias acorde a las necesidades y requerimientos surgidos en el momento.
- Implementar el sistema desarrollado en los respectivos equipos a usar siendo estos: servidor y equipos clientes.
- Capacitar a las usuarias finales en el manejo adecuado del sistema tanto configuración del Sistema Administrador y Base de Datos como en el programa de usuario.

#### **1.4. ALCANCE Y LIMITACIONES**

El proyecto a desarrollar será de uso exclusivo del Centro de Difusión y

Documentación del Consejo Nacional de las Mujeres además el personal del área de informática tendrá acceso al sistema y a su código fuente para las respectivas modificaciones o mantenimientos.

Cabe recalcar que la herramienta a desarrollar es para un Centro de Documentación y Difusión especializada en género (femenino) más no para una biblioteca, el centro de documentación y difusión el cual a más de aportar con material bibliográfico de lectura, apoya con materiales de donación a diversas organizaciones de mujeres.

Como funciones y módulos a desarrollar están los siguientes:

Crear un módulo para poder llevar un adecuado control sobre el inventario de los materiales a difundir y de donación, para lo cual se controlará la información que se manipule (ingresar, modificar, eliminar) a través de procesos y además de información almacenada en las respectivas tablas.

- **Funcionaria:** almacenará información básica sobre las encargadas del centro de difusión y documentación.
- **Tipo Uso de Material:** almacenará información básica sobre los tipos de uso que van a tener los materiales (biblioteca, charla, conferencia, consumo interno, difusión, estudios, eventos nacionales, eventos internacionales, foros, investigación, consultoría, etc.).
- **Tipos de Unidad:** los tipos de unidad son la presentación que tienen los materiales a difundir pueden ser docenas, medias docenas, centenas, unidades, etc.
- **Formato de tipo de documento:** se almacenará el formato del documento a usarse para las transacciones, pueden ser hojas de ruta, solicitud de entrega – recepción, email, verbal, documentos digitales, entre otros.
- **Productos:** Se almacenará los datos de los productos o materiales a difundir, tendrá un código, una descripción, el costo, stock disponible y una alarma la cual dirá cuando un material está pronto a agotarse.
- **Unidades Operativas (Ubicación):** Se pone el país, provincia o lugar a donde va el material

- **Beneficiarios x Ubicación (Responsable):** Se almacenará la información necesaria sobre el responsable o quien recibe el material difundido o donado.
- **Usuario Final:** Se almacenará información necesaria sobre el usuario final, estos pueden ser: colegios, comités de usuarias, embajadas, empresas, escuelas, universidades, estado con sus diversas posibilidades, librerías, medios de comunicación, organizaciones de mujeres, personal de la institución, etc.

Registro de los materiales de difusión y donación, se implementará las siguientes transacciones:

- **Ingreso:** Se registra los materiales que ingresan a stock o bodega.
- **Egresos:** Se registra los egresos o salidas de bodegas sobre el material que se envíe sean estos donados, difundidos o devueltos, según la opción que se tome.

Dotar de reportes estadísticos periódicos, estos reportes podrán ser creados a necesidad y propósito de cada usuario o circunstancia, los reportes a presentar constarán de datos estadísticos según el tipo de reporte que se escoja, también se podrá elegir el tipo de gráfico estadístico a presentar, además estos reportes ayudarán a desarrollar una mejor planificación y proyección sobre los materiales de difusión y donación, los reportes que serán mostrados son los siguientes:

- Reportes de Stock Disponible
- Reportes para resumen de Ingresos
- Ingresos Totales
- Objeto Receptor o usuario final
- Egresos Totales
- Egresos por Tipo de Transacción
- Acumulado por Producto
- Movimientos de Producto
- Movimientos por Destinatario

Se deberá desarrollar y adecuar un módulo de Planificación sobre los materiales

de difusión y donación, el cuál apoye a las funcionarias del área a realizar con anticipación pedidos también tener información necesaria sobre valores planificados y valores usados o ejecutados en el área de documentación y difusión, esta información junto con los reportes que se obtiene en el sistema serán usados por las encargadas del área para realizar una adecuada planificación de las actividades y los valores a usarse en el siguiente año.

El módulo deberá llevar un seguimiento de las actividades, valores planificados y ejecutados en la planificación del área también se ingresaran los objetivos, metas y las actividades con sus respectivos valores, responsables y contrapartes.

Además el módulo permitirá manejar un listado con datos personales de las instituciones que aportan con material al CONAMU, para poder en un futuro saber el apoyo obtenido a lo largo de un año o un determinado tiempo.

Se desarrollará un módulo para poder obtener reportes con datos, gráficos y análisis estadísticos obtenidos con la información almacenada mediante consultas personalizadas, los reportes detallarán la información mediante indicadores, valores, porcentajes, gráficos estadísticos a elegir por la funcionaria, el módulo aportará a las encargadas del área una mayor apreciación de la situación de los materiales donados y difundidos, destinatarios finales, materiales de mayor y menor demanda, sitios geográficos de mayor y menor envío, organizaciones de mujeres con mayor adquisición de materiales, temas de consulta de mayor demanda. Los reportes tendrán la facilidad de ser exportados a formatos; Excel, Word, pdf.

El módulo contará con una encuesta virtual, el cuál será llenado por los visitantes al centro de documentación y difusión, la encuesta se la hará con la finalidad de obtener información de cómo se lleva el área, condición y actualidad de los materiales, atención de las funcionarias, temas de mayor consulta, entidades u organizaciones de mayor visita al área.

El manual de usuario será entregado a las encargadas del área de difusión y documentación también se entregará una copia del manual al departamento informático, el manual de usuario permitirá usar de una manera fácil el sistema.

También se entregará un manual técnico o de mantenimiento y adicionalmente el diccionario de datos al departamento informático que permitirá a las personas encargadas de sistemas realizar mantenimientos, actualizaciones y adecuaciones necesarias acorde a las necesidades y requerimientos surgidos en el momento sobre el sistema y la base de datos.

El proyecto estará implementado sobre plataformas que manejen software libre y que estén a disponibilidad del Consejo Nacional de las Mujeres:

- **Apache** es un servidor http de software libre, es de código abierto para plataformas Linux (Unix), Windows, Macintosh y otras, que implementa el protocolo http (HyperText Transfer Protocol).

En este servidor será donde se alojarán las páginas desarrolladas y las funciones de servidor.

- **JavaScript** es un lenguaje interpretado orientado a las páginas Web, con una sintaxis semejante a la del lenguaje Java. Se utiliza en páginas Web HTML, para realizar tareas y operaciones en el marco de la aplicación cliente.

Bajo este lenguaje se realizará el diseño del menú y el control de registro de datos del sistema.

- **PHP** es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas Web dinámicas. Es usado principalmente en interpretación del lado del servidor pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica.

Mediante el diseño de páginas php se alojarán las aplicaciones realizadas que serán posteriormente visualizadas por las usuarias.

## 1.5. JUSTIFICACIÓN

Posibilitar el control del proceso de adquisición y distribución del material a ser entregado a las diferentes organizaciones de mujeres a través de esta herramienta de software.

Con el resultado de la implementación de esta herramienta y adicionalmente con la dotación de reportes que facilite gráficos estadísticos que permita el registro de adquisición de materiales, movimiento y distribución el Centro de Difusión y Documentación aumentará su eficiencia en la ejecución de proyecciones, donaciones, difusiones y planificación adecuada del material a facilitar en beneficio de las diferentes organizaciones de mujeres en el territorio nacional.

## **1.6. SITUACIÓN ACTUAL**

El área del Centro de Difusión y Documentación del Consejo Nacional de las mujeres en la actualidad realiza todas sus funciones y movimientos mediante el Sistema Contable Financiero llamado OLYMPO desarrollado por la empresa PROTECOTELSA, el cuál permite entre sus funciones los ingresos y egresos de materiales mas no un control adecuado de los mismos, este sistema no satisface los requerimientos específicos del área antes mencionada ya que no dispone de funciones como manejo de características de los materiales de difusión y donación, administración de ubicaciones o destino final, responsables y usuario final, el manejo en transacciones de ingreso tanto para compra como para devolución, los reportes que se obtiene mediante el sistema OLYMPO no satisfacen las necesidades de las encargadas del área, mediante el sistema OLYMPO los reportes útiles son de stock disponible y resúmenes de ingresos y egresos, a más de estos reportes se necesitan de reportes como: acumulados por producto, movimientos de producto, movimientos por destinatario, egresos por tipo de transacción, egresos por objeto receptor(usuario final).

Por esta razón es imprescindible dotarle al área de Difusión y Documentación de un sistema que le ayude a controlar cada una de las actividades que realiza de acuerdo a sus necesidades específicas.

## **1.7. SOLUCIÓN PROPUESTA**

Es necesario obtener información sobre utilitarios para el diseño, desarrollo e implementación de la base de datos, interfaces, módulos y procesos necesarios para el adecuado desempeño del sistema.

El sistema que se va a desarrollar contará al inicio de su ejecución con un interfaz de seguridad en cual verificará lo datos ingresados tanto como usuario y contraseña, con lo cual las encargadas de manipular el sistema ingresaran al mismo con sus respectivos privilegios de cuenta, al ingresar a su respectiva cuenta el sistema visualizará un menú de opciones dependiendo de los privilegios de cada usuario sea administrador, operador y digitador.

El sistema a entregar será desarrollado con herramientas de software libre como el PostgreSQL para el diseño y desarrollo de la base de datos por su extensa funcionalidad, seguridad, administración de información y gran capacidad de almacenamiento que posee, el PHP para la creación de la interfaz, desarrollo de procesos y funciones del sistema, el lenguaje de programación PHP es de libre acceso y contiene muchas funciones para seguridad, creación de funciones y procesos, interacción con base de datos, todo lo mencionado a nuestro parecer es ideal para el desarrollo de páginas Web dinámicas funcionales.

Por estas razones la herramienta informática a desarrollar no se sujetará a un solo sistema operativo, pues puede ser configurado y operar sin problemas tanto en Linux como en Windows. Cabe mencionar que el sistema para ser ejecutado en las terminales necesitará únicamente tener un explorador de Internet sea este Internet Explorer, Firefox Mozilla, Safari, Opera, entre otros.

## **1.8. METODOLOGÍA**

La metodología que se utilizará en el desarrollo de la propuesta para la generación de la herramienta informática a entregar en el Centro de Difusión y Documentación del Consejo Nacional de las Mujeres es la Metodología XP debido a que esta metodología se sujeta mas a nuestras necesidades y expectativas en el desarrollo de nuestro proyecto.

La metodología de Programación Extrema (Extreme Programming) usualmente conocida como XP es un enfoque de la ingeniería de software formulado por Kent Beck en 1996. Es la más destacada de los procesos ágiles de desarrollo de software. Se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad.



La metodología XP se usa para el desarrollo de sistemas a corto plazo, con un equipo no tan numeroso y cuya necesidad en la entrega del sistema es casi inmediata. Esta metodología consiste en una programación ágil, en el proceso de desarrollo se tiene como parte del equipo al usuario final pues es una parte esencial en el éxito de la culminación del proyecto.

Esta metodología tiene como aspectos fundamentales:

- La comunicación entre usuarios y los desarrolladores
- Simplicidad al desarrollar y codificar los módulos del sistema
- La retroalimentación concreta y frecuente del equipo de desarrollo, el cliente y los usuarios finales.

La metodología XP se basa en la idea de que existen cuatro variables que guían el desarrollo de sistemas<sup>1</sup>:

Coste: (máquinas, especialistas y oficinas).

Tiempo: (total y de entrega).

Calidad: (externa e interna).

Alcance (intervención del cliente).

Donde la relación entre estas variables no es directa, y a más calidad menos tiempo.

Esta metodología consta de cuatro valores:

Comunicación: Mantener un buen canal de comunicación entre las personas del grupo de trabajo.

Sencillez: Optar por usar lo más simple que pueda funcionar y que se pueda utilizar a futuro.

Retroalimentación: Por medio de pruebas funcionales realizadas al sistema este informará el grado de fiabilidad del sistema.

Valentía: Asumir retos, ser valiente ante los problemas y afrontarlos.

El uso de metodología XP surgió como repuesta y posible solución a los problemas derivados del cambio en los requerimientos.

---

<sup>1</sup> Metodología XP. [en línea]. <<http://brconsulting.info/portal/articulos/metodologias-de-desarrollo/extreme-programming---xp.html>>

Fases de la Metodología XP<sup>2</sup>:

### **- Planificación**

XP plantea la planificación como un permanente diálogo entre las partes, la empresarial y la técnica. Las personas del negocio necesitan determinar:

Historia de Usuario: Son las técnicas utilizadas en XP para especificar los requisitos del software. Se trata de las necesidades escritas, en donde el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales.

Plan de Entrega: Estima el tiempo de desarrollo de las historias de usuario.

Iteraciones: Identificar las historias de usuario que se van a desarrollar en una iteración específica.

Rotaciones: Permite que todas las personas conozcan cómo funciona el sistema.

Reuniones: Con el fin de facilitar la comunicación entre el grupo de trabajo y la exposición de los diferentes problemas se sugiere reuniones diarias.

Correcciones: Se debe corregir el proceso cuando éste falle.

Para la planificación del sistema se realizó reuniones con las funcionarias para acordar los requisitos que el software debe resolver para que éste genere valor, especificando que es lo que se debe hacer en primer lugar. Además se coordinó sobre las fechas aproximadas para las primeras versiones y los avances que el sistema iría aportando a la Institución. Se informó a las funcionarias sobre las consecuencias de la toma de decisiones en el caso que se quisiera realizar modificaciones ó cambios ( base de datos), ya que esto implicaría tiempo.

### **- Diseño**

La metodología XP sugiere que hay que conseguir diseños simples y sencillos. Hay que procurar hacer lo menos complicado posible para conseguir un diseño fácilmente entendible y con posibilidad de implementar y que a un futuro signifique menos tiempo y esfuerzo a desarrollar.

Metáfora de sistema: La tarea de elegir una metáfora para el sistema que permita mantener la coherencia de nombres de todo aquello que se va implementar. Además que ayude a cualquier persona entender el objeto del programa.

Soluciones: Se propone soluciones a problemas técnicos o de diseño.

---

<sup>2</sup> Metodología XP. [en línea].< <http://programacionextrema.tripod.com/fases.htm>>

Funcionalidad mínima: Se ignora las funcionalidades extras que podrían incorporarse al proyecto, se trata de centrar en lo principal.

Riesgos: Si surge algún problema durante el diseño se sugiere que los desarrolladores investiguen y reduzcan al máximo el riesgo que supone el problema.

Reciclaje: Implicará en mantener el código limpio y fácil de comprender, modificar y ampliar.

Para la realización del diseño del sistema se optó por la forma más sencilla posible, tanto en el diseño de base de datos como en el sistema, en el que se trató de manejar de una forma específica los nombres de los componentes del sistema. De esta manera entregar un sistema satisfactorio a las funcionarias.

#### **- Codificación**

La codificación debe ser realizada atendiendo a estándares de codificación ya creados. Programar bajo estándares mantiene el código consistente y facilita su comprensión y escalabilidad.

Programación por parejas: Permite obtener un código más eficiente e incrementa la productividad y la calidad del software.

Se realizó la codificación para el sistema manejando los estándares sugeridos y se logró un código entendible, capaz de ser modificado o implementado si la Institución lo necesitara. El código fuente comentado se muestra en el Anexo B.

#### **- Pruebas**

No debe existir ninguna característica en el programa que no haya sido probada, los programadores escriben pruebas para verificar el correcto funcionamiento del programa, los clientes realizan pruebas funcionales. Se tiene como resultado un programa más seguro que conforme pasa el tiempo es capaz de aceptar nuevos cambios. Se crea las pruebas a partir de las historias de usuario. Una historia de usuario no se considera lista hasta que haya pasado todas sus pruebas y el cliente es el que evalúa.

En el sistema que se ha desarrollado se procedió a realizar las pruebas para validar el funcionamiento y consistencia de la aplicación. Se realizó las pruebas de caja negra, de plataforma y de funcionalidad con la intervención de las funcionarias de la Institución. Las pruebas realizadas se muestra en el Capítulo 7.

## **1.9. HARDWARE Y SOFTWARE A UTILIZAR**

Los requerimientos básicos y necesarios para utilizar el sistema a desarrollar e implementar tanto en el servidor como en las PC de cada funcionaria encargada del manejo del sistema son los siguientes:

### **Para Servidor:**

- Sistema Operativo Linux (puede ser cualquier plataforma)
- Sistema Operativo Windows 2000 Server
- Servidor Web – Apache 4.5
- Servidor de Base de Datos – Postgre SQL 8.4.0

### **Requisitos mínimos del sistema**

- PC con procesador Intel Pentium III a 500 MHz o superior
- 512 MB de memoria RAM
- 8 GB de espacio de disco disponible
- Pantalla con resolución 800 x 600

### **Configuración recomendada**

- PC con procesador Intel Pentium IV 1.8 MHz
- 1 GB de memoria RAM
- 40 GB de espacio de disco disponible
- Pantalla con resolución 1024 x 768

### **Para Terminal:**

- Sistema Operativo Windows 98 o superior
- Navegador de Internet: Internet Explorer, Netscape Navigator, Firefox Mozilla.

### **Requisitos mínimos del sistema**

- PC con procesador Intel Pentium III de 500Mhz o superior
- 256 Mb de memoria RAM
- 4 GB de espacio de disco disponible
- Pantalla con resolución 800 x 600 o superior

## CAPÍTULO 2

### 2. MARCO TEÓRICO

#### 2.1. INTRODUCCIÓN A PHP

A PHP se le presentó con el nombre “Pre-Procesador Hipertexto” (*Hypertext Pre-Processor*). PHP es un preprocesador del hipertexto y por ende, se ejecuta en un servidor Web remoto para procesar páginas Web antes de que sean cargadas en el navegador.

Además de sus potentes características, PHP, es un lenguaje simple que ha sido diseñado específicamente para el desarrollo y la producción de páginas Web<sup>3</sup>.

PHP es un software de código abierto (*open-source*) y puede ser descargado gratuitamente.

El lenguaje de programación PHP es uno de los lenguajes más extendidos en el Internet y que se ejecuta en el lado de los servidores Web. PHP es un lenguaje de desarrollo Web creciente y ha tenido una gran aceptación en la comunidad de desarrolladores de páginas Web debido sobre todo a la potencia y simplicidad que lo caracterizan.

PHP permite crear contenido dinámico en las páginas HTML y sobre todo algo muy importante para el desarrollo de los proyectos, es que PHP ofrece un sin fin de funciones para la explotación de bases de datos de una manera llana, sin complicaciones.

PHP dispone de miles de herramientas que permiten acceder a bases de datos de forma sencilla, por lo que es ideal para crear varias y diversas aplicaciones para Internet.

El código generado es mantenible<sup>4</sup>, se lee muy bien y se puede programar utilizando objetos. PHP es un lenguaje fácil de utilizar por lo tanto es ideal para el que comienza a desarrollar aplicaciones Web como para el desarrollador experimentado.

---

<sup>3</sup> De la Cruz Villar, Joel. *PHP y MySQL*. Lima: Megabyte, 2006, p. 3.

<sup>4</sup> Código mantenible es puramente código que puede mantenerse durante un ciclo de tiempo y en diferentes plataformas.

También se puede mencionar que PHP es multiplataforma, funciona tanto para Linux como para Windows de forma que el código que se haya creado para una de ellas no tiene porque codificarse al pasar a la otra.

La sintaxis que utiliza es similar a la de C y Perl. Para quien conozca el lenguaje ASP (*Active Server Pages*) de Microsoft, PHP es muy parecido a diferencia que es gratuito y multiplataforma. PHP es conocido por sus características avanzadas, a continuación se menciona las características fundamentales.

- Es un lenguaje de script de código abierto para servidores.
- Es independiente del sistema operativo y puede ser utilizado en cualquiera de ellos, incluyendo Microsoft Windows, Mac OS, Linux y Solaris, entre otros.
- Utiliza una amplia gama de servidores Web, útiles como Apache, Microsoft Internet, Information Server, Netscape e iPlanet.
- Se conecta a gran cantidad de bases de datos, como MySQL, Ingres, Sybase, Oracle, PostgreSQL, Informix, FrontBase y Unix dbm.
- El código de PHP es más simple que en otros lenguajes de script<sup>5</sup>.

El funcionamiento de PHP es bastante simple:

- Se escribe o desarrolla las páginas HTML pero con el código PHP dentro.
- El código PHP empieza con **<?php** y termina con **?>**, las instrucciones finalizan con punto y coma ( ; ) y las variables se identifican con el signo de dólar (\$). Se puede insertar código donde se desee y cuantas veces sea necesario.
- Se guarda la página en el servidor Web.
- Un navegador solicita una página al servidor.
- El servidor interpreta el código PHP.

El servidor envía el resultado del conjunto de código HTML y el resultado del código PHP que también es HTML

---

<sup>5</sup> De la Cruz Villar, Joel. Opt. Cit. p. 3-4.

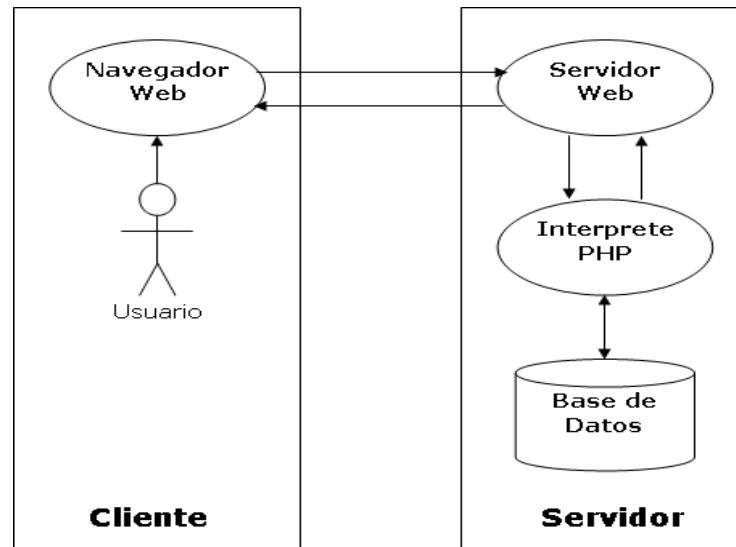


Figura 2.1. Esquema de solicitud de página al servidor. Los Autores.

### 2.1.1. VARIABLES

En PHP las variables se representan como un signo de dólar seguido por el nombre de la variable. En nombre de la variable es sensible a minúsculas y mayúsculas<sup>6</sup>. Una variable es un contenedor de información, en PHP una variable se distingue porque va antecedido de el signo de dólares (\$), por ejemplo \$a, \$b, \$pais, etc.

Una variable puede contener; cadenas, enteros, decimales o flotantes, arrays, booleanos, objetos.

- **Cadenas:** PHP dispone de toda una gama de funciones que permiten trabajar cómodamente con las cadenas. En PHP existe una interpretación diferente entre las comillas simples y las comillas dobles. Si la cadena esta entre comillas dobles las variables que estén dentro de la cadena serán expandidas. Por ejemplo:

```

$a = "Mundo";
echo ' Hola $a '; //Esta línea imprimirá en pantalla Hola $a
echo " Hola $a "; //Esta línea imprimirá en pantalla Hola Mundo
  
```

Las cadenas se pueden concatenar usando el operador punto (.).

Por ejemplo:

<sup>6</sup> De la Cruz Villar, Joel. Opt. Cit. p. 32.

```
$a = "Hola";
$b =$a . "Mundo"; //Ahora la variable $b contiene "Hola Mundo"
```

- **Enteros:** Un valor entero o integer es un número del conjunto  $Z = \{..., -2, -1, 0, 1, 2, ...\}$ . El tamaño de un entero es dependiente de la plataforma, aunque un valor máximo de aproximadamente dos billones es el valor usual (32 bits con signo). PHP no soporta enteros sin signo. Los enteros se pueden especificar usando una de las siguientes sintaxis. Ejemplo:

```
<?php
$a = 1234; //número decimal
$a = -123; //un número negativo
$a = 0123; //número octal (equivalente al 83 decimal)
$a = 0x12; //número hexadecimal (equivalente al 18 decimal)
?>
```

- **Flotantes:** Los números de punto flotante o llamados también doubles, flotantes o números reales, se pueden especificar utilizando cualquiera de las siguientes sintaxis.

```
<?php
$a = 1.234;
$b = 1.2e3;
$c = 7E-10;
?>
```

- **Arrays:** Una matriz en PHP es en realidad un mapa ordenado. Un mapa es un tipo de datos que asocia valores con claves. Este tipo es optimizado en varias formas, de modo que puede usarlo como una matriz real, o una lista (vector).
  - **Arrays Unidimensionales:** Se puede crear un array simplemente añadiendo valores al array. Cuando se asigna un valor a una variable array usando corchetes vacíos, el valor se añadirá al final del array. Ejemplo:

```
<?php
$a[ ] = "hola"; // $a[2] == "hola"
$a[ ] = "mundo"; // $a[3] == "mundo"
?>
```



- **Arrays Multidimensionales:** Para cada dimensión del array, se puede añadir otro valor[clave] al final. Ejemplo:

```
<?php
    $A[1] = $f; //una sola dimensión
    $A[1][0] = $f; //bidimensional;
    $A["foo"][4]["bar"][0] = $f; //tetradimensional
?>
```

- **Booleanos:** Un booleano expresa un valor de verdad. Para especificar un literal booleano, se usa alguna de las palabras clave **TRUE** o **ALSE**.

```
<?php
    $foo = True; //asignar el valor de TRUE a $foo
?>
```

- **Objetos:** Una variable object se usa para instanciar un objeto a una variable, para inicializar un objeto se usa la sentencia new. Ejemplo:

```
<?php
    class foo{
        function hacer_foo(){
            echo "Haciendo foo.";
        }
    }

    $bar = new foo;
    $bar->hacer_foo();
?>
```

Dentro de las variables en especial las variables tipo texto o string, se puede hacer uso de varios caracteres especiales que ayudarán a la depuración del código, estos caracteres son los siguientes:

- **\"** : Carácter doble comillas.
- **\\** : Carácter barra invertida.
- **\n** : Nueva línea.
- **\r** : Retorno de carro.
- **\t** : Tabulador horizontal

En PHP existe dos tipos de variables; las variables locales que pueden ser usadas dentro de una misma función, y las variables globales que pueden ser usadas fuera de sus funciones, para acceder a estas variables globales desde una función se usa la instrucción *global*/ nombre\_variable.

Las constantes son similares a una variable con la diferencia que no lleva el signo de dólar antes de la variable y que solo se define su valor una sola vez, para definir una constante se usa la instrucción define, por ejemplo:

```
define ("Nombre_Constante", "Valor");
printf (Nombre_Constante);
```

## 2.1.2. OPERADORES

Un operador permite crear, modificar y comparar variables. Mediante un operador se puede obtener un valor a partir de usar uno o mas valores que estén contenidas en variables.

### 2.1.2.1 Precedencia de operadores

La precedencia de un operador indica que tan cerca esta agrupa dos expresiones. Por ejemplo en la expresión  $1 + 5 * 3$  el resultado que arroja es 16 y no 18, puesto que el operador de multiplicación "\*" tiene una mayor precedencia que el operador de adición "+". Los paréntesis pueden ser usados para marcar la precedencia, si es necesario. Por ejemplo:  $(1 + 5) * 3$  que arroja como resultado 18. Si la precedencia de los operadores es la misma, se utiliza una agrupación de izquierda a derecha.

Asociatividad	Operadores
derecha	[
derecha	! ~ ++ -- (int) (float) (string) (array) (object) @
izquierda	* / %
izquierda	+ - .
izquierda	<< >>
no-asociativos	< <= > >=
no-asociativos	== != === !==
izquierda	&
izquierda	^
izquierda	
izquierda	&&
izquierda	
izquierda	? :
derecha	= += -= *= /= .= %= &=  = ^= <<= >>=
derecha	print
izquierda	and
izquierda	xor
izquierda	or
izquierda	,

Tabla 2.1. Precedencia de Operadores

### 2.1.2.2. Operadores aritméticos

Los operadores aritméticos son los más fáciles de aplicar dentro de un código. Estos operadores permiten realizar operaciones numéricas con las variables.

Nombre	Ejemplo	Resultado
Adición	$\$a + \$b$	Suma de $\$a$ y $\$b$ .
Substracción	$\$a - \$b$	Diferencia entre $\$a$ y $\$b$ .
Multiplicación	$\$a * \$b$	Producto de $\$a$ y $\$b$ .
División	$\$a / \$b$	Cociente de $\$a$ y $\$b$ .
Módulo	$\$a \% \$b$	Resto de $\$a$ dividido por $\$b$ .

*Tabla 2.2. Operadores Aritméticos*

El operador de división (/) devuelve un valor flotante en todos los casos, incluso si las dos variables u operandos son enteros (o cadenas que son convertidas a enteros)<sup>7</sup>.

### 2.1.2.3. Operadores de asignación

Los operadores de asignación sirven para asignar un valor a una variable, la variable de la izquierda recibe el valor de la expresión de la derecha.

El valor de una expresión de asignación es el valor que se asigna. Es decir, el valor de “ $\$a = 3$ ” es 3<sup>8</sup>.

En conjunto con el operador básico de asignación (=), existen operadores combinados para todas las operaciones de aritmética binaria y de cadenas, que permiten usar un valor en una expresión y luego definir su valor como el resultado de esa expresión. Por ejemplo:

```
<?php
    $a = 3;
    $a += 5;
    //Define $a es 8, es lo mismo que $a = $a + 5;
    $b = "Hola";
    $b .= " a todos";
    //Define $b como "Hola a todos", tal como $b = $b . "a todos";
?>
```

<sup>7</sup> De la Cruz Villar, Joel. *PHP y MySQL*. Lima: Megabyte, 2006, p. 48.

<sup>8</sup> De la Cruz Villar, Joel. *Opt. Cit.* p. 50.

Ejemplo	Descripción
\$a = \$b;	Asigna a \$a el contenido de \$b
\$a += \$b;	Le suma a \$b a \$a
\$a -= \$b;	Le resta a \$b a \$a
\$a *= \$b;	Multiplica \$a por \$b y lo asigna a \$a
\$a /= \$b;	Divide \$a por \$b y lo asigna a \$a
\$a .= \$b;	Añade la cadena \$b a la cadena \$a

Tabla 2.3. Operadores de Asignación

#### 2.1.2.4. Operadores bit a bit

Los operadores bit<sup>9</sup> a bit le permiten activar o desactivar bits individuales de un entero. Si los parámetros tanto a la izquierda y a la derecha son cadenas, el operador bit a bit trabajará sobre los valores ASCII<sup>10</sup> de los caracteres. Por Ejemplo:

```
<?php
echo 12 ^9; //imprime 5
echo "12" ^"9"; //imprime el carácter Backspace (ascii 8)
                // ("1" (ascii 49)) ^("9" (ascii 57)) = #8
echo "hallo" ^"hello"; //imprime valores ascii #0 #4 #0 #0 #0
                //"a" ^"e" = #4
?>
```

Ejemplo	Nombre	Resultado
\$a & \$b	Y	Los bits que están activos tanto en \$a como en \$b son activados
\$a   \$b	O	Los bits que están activos ya sea en \$a o en \$b son activados
\$a ^ \$b	O exclusivo (Xor)	Los bits que estén activos en \$a o \$b, pero no en ambos son activados
~ \$a	No	Los bits que estén activos en \$a son desactivados y viceversa
\$a << \$b	Desplazamiento a izquierda	Desplaza los bits de \$a, \$b pasos a la izquierda (cada paso quiere decir "multiplicar por dos")
\$a >> \$b	Desplazamiento a derecha	Desplaza los bits de \$a, \$b pasos a la derecha (cada paso quiere decir "dividir por dos")

Tabla 2.4. Operadores BIT a BIT

#### 2.1.2.5. Operadores de comparación

Los operadores de comparación en su mayoría binarios, permiten

<sup>9</sup> Bit es la unidad mínima de información que puede almacenar y manejar un ordenador, equivalente a un 0 o un 1.

<sup>10</sup> American Standard Code for Information Interchange, un código de 7-bit que sustituye las letras del alfabeto romano por cifras y otros caracteres informáticos.

comparar dos valores, devolviendo un valor booleano 1(true) si la condición se cumple y 0(false) si la condición no se cumple.

Ejemplo	Nombre	Resultado
\$a == \$b	Igual TRUE	si \$a es igual a \$b
\$a === \$b	Idéntico TRUE	si \$a es igual a \$b y son del mismo tipo ( en PHP 4)
\$a != \$b	Diferente TRUE	si \$a no es igual a \$b.
\$a <> \$b	Diferente TRUE	si \$a no es igual a \$b.
\$a !== \$b	No idénticos TRUE	si \$a no es igual a \$b, o si no son del mismo tipo. ( en PHP 4)
\$a < \$b	Menor que TRUE	si \$a es estrictamente menor que \$b.
\$a > \$b	Mayor que TRUE	si \$a es estrictamente mayor que \$b.
\$a <= \$b	Menor o igual que TRUE	si \$a es menor o igual que \$b.
\$a >= \$b	Mayor o igual que TRUE	si \$a es mayor o igual que \$b.

Tabla 2.5. Operadores de Comparación

Otro operador condicional es el operador “?:” (o ternario)<sup>11</sup>.

```
<?php
// Ejemplo de uso de: el Operador Ternario
$accion = (empty($_POST['accion'])) ? 'predeterminada' :
$_POST['accion'];
// La sentencia anterior es idéntica a este bloque if/else
if (empty($_POST['accion'])) {
    $accion = 'predeterminada';
} else {
    $accion = $_POST['accion'];
}
?>
```

#### 2.1.2.6. Operadores de control de errores

PHP ofrece soporte para un operador de control de errores: el signo de arroba (@). Cuando es colocado al comienzo de una expresión en PHP, cualquier mensaje de error que pudiera generarse a causa de esa expresión será ignorado. Si la característica `track_errors` está habilitada, cualquier mensaje de error generado por la expresión será almacenado en la variable `$php_errormsg`. La variable será sobrescrita en cada instancia de error, entonces hay que realizar los chequeos de forma temprana si se desea usarla<sup>12</sup>. Ejemplo:

<sup>11</sup> De la Cruz Villar, Joel. *PHP y MySQL*. Lima: Megabyte, 2006, p. 52.

<sup>12</sup> De la Cruz Villar, Joel. Op. Cit. p. 52 - 53.

```

<?php
/* Error intencional de archivo */
$mi_archivo = @file ('archivo_que_no_existe') or
die ("La apertura de archivo ha fallado: el error fue
'$php_errormsg'");
// Esto funciona con cualquier expresión, no solo con funciones:
$valor = @$cache[$llave];
// No producirá una anotación si el índice $llave no existe.
?>

```

Nota: El operador @ trabaja sólo sobre expresiones. Hay que tener como regla de oro lo siguiente: si se puede tomar el valor de algo, entonces se puede usar el operador @ sobre ese algo. Por ejemplo, se lo puede usar al inicio de variables, llamadas a funciones y sentencias include(), constantes, y así sucesivamente. No se lo puede usar sobre definiciones de función o clase, ni sobre estructuras condicionales como if y foreach, y así sucesivamente.

Nota: El operador de prefijo "@" para control de errores no deshabilitará los mensajes que son resultado de errores en la fase de análisis sintáctico.

Aviso: En la actualidad, el operador de prefijo "@" para control de errores deshabilitará incluso el reporte de errores en casos de fallos críticos que terminarán la ejecución del script. Entre otras cosas, esto quiere decir que si usa "@" para eliminar los errores de una cierta función, y ésta no se encuentra disponible o ha sido escrita de forma incorrecta, el script se detendrá en ese punto sin dar indicación alguna del motivo<sup>13</sup>.

#### 2.1.2.7. Operadores de ejecución

PHP soporta un operador de ejecución: las comillas invertidas (`). PHP intentará ejecutar el contenido entre las comillas como si se tratara de un comando del intérprete de comandos; su salida será devuelta (es decir, no será simplemente volcada como salida; puede ser asignada a una variable). El uso del operador de comillas invertidas es idéntico al de shell\_exec()<sup>14</sup>. Ejemplo:

<sup>13</sup> De la Cruz Villar, Joel. *PHP y MySQL*. Lima: Megabyte, 2006, p. 54.

<sup>14</sup> De la Cruz Villar, Joel. Op. Cit. p. 54.

```
<?php
    $salida = `ls -al`;
    echo "<pre>$salida</pre>";
    ?>
```

El operador de comillas invertidas es deshabilitado cuando se encuentra activo safe mode o cuando se deshabilita shell\_exec().

### 2.1.2.8. Operadores de incremento / decremento

PHP ofrece soporte de operadores de pre y post incremento y decremento estilo C<sup>15</sup>.

Ejemplo	Nombre	Efecto
++\$a	Pre – incremento	Incrementa \$a en uno y luego devuelve \$a.
\$a++	Post – incremento	Devuelve \$a y luego incrementa \$a en uno.
--\$a	Pre – decremento	Decrementa \$a en uno, luego devuelve \$a.
\$a--	Post – decremento	Devuelve \$a, luego decrementa \$a en uno.

Tabla 2.6. Operadores de Incremento / Decremento

Ejemplo:

```
<?php
echo "<h3>Postincremento</h3>";
$a = 5;
echo "Debe ser 5: " . $a++ . "<br />\n";
echo "Debe ser 6: " . $a . "<br />\n";
echo "<h3>Preincremento</h3>";
$a = 5;
echo "Debe ser 6: " . ++$a . "<br />\n";
echo "Debe ser 6: " . $a . "<br />\n";
echo "<h3>Postdecremento</h3>";
$a = 5;
echo "Debe ser 5: " . $a-- . "<br />\n";
echo "Debe ser 4: " . $a . "<br />\n";
echo "<h3>Predecremento</h3>";
$a = 5;
echo "Debe ser 4: " . --$a . "<br />\n";
echo "Debe ser 4: " . $a . "<br />\n";
?>
```

### 2.1.2.9. Operadores de cadena

Existen dos operadores para datos tipo string.

El primero es el operador de concatenación ('.'), el cual devuelve el resultado de concatenar sus argumentos a lado derecho e izquierdo.

<sup>15</sup> De la Cruz Villar, Joel. Op. Cit. p. 55.

El segundo es el operador de asignación sobre concatenación ('.='), el cual adiciona el argumento del lado derecho al argumento en el lado izquierdo<sup>16</sup>.

Ejemplo:

```
<?php
$a = "&iexcl;Hola ";
$b = $a . "Mundo!"; // ahora $b contiene "&iexcl;Hola Mundo!"
$a = "&iexcl;Hola ";
$a .= "Mundo!"; // ahora $a contiene "&iexcl;Hola Mundo!"
?>
```

#### 2.1.2.10. Operadores de tipo

PHP tiene un operador único de tipo: instanceof, es usado para determinar si un objeto dado es de una clase de objeto especificada.

El operador instanceof fue introducido en PHP 5. Ejemplo:

```
<?php
class A { }
class B { }
$cosa = new A;
if ($cosa instanceof A) {
    echo 'A';
}
if ($cosa instanceof B) {
    echo 'B';
}
// Imprimirá A
?>
```

Dado que \$cosa es un object de tipo A, pero no de tipo B, sólo el bloque dependiente del tipo A será ejecutado.

#### 2.1.2.11. Operadores lógicos

Se usan en combinaciones con los operadores de comparación cuando la expresión de la condición lo requiere.

Ejemplo	Nombre	Resultado
\$a and \$b	Y	TRUE si tanto \$a como \$b son TRUE
\$a or \$b	O	TRUE si cualquiera de \$a o \$b es TRUE
\$a xor \$b	O exclusivo (Xor)	TRUE si \$a o \$b es TRUE, pero no ambos
! \$a	No	TRUE si \$a no es TRUE
\$a && \$b	Y	TRUE si tanto \$a como \$b son TRUE
\$a    \$b	O	TRUE si cualquiera de \$a o \$b es TRUE

Tabla 2.7. Operadores Lógicos

<sup>16</sup> De la Cruz Villar, Joel. Op. Cit. p. 57.



### 2.1.3. ESTRUCTURAS DE CONTROL

Todo script PHP se compone de una serie de sentencias. Una sentencia puede ser una asignación, una llamada a función, un bucle, una sentencia condicional e incluso una sentencia que no haga nada (una sentencia vacía).

Las sentencias normalmente acaban con punto y coma. Además, las sentencias se pueden agrupar en grupos de sentencias encapsulando un grupo de sentencias con llaves. Un grupo de sentencias es también una sentencia<sup>17</sup>.

#### 2.1.3.1. If

La construcción if es una de las más importantes características de muchos lenguajes, incluido PHP. Permite la ejecución condicional de fragmentos de código. PHP caracteriza una estructura if que es similar a la de C. Ejemplo:

```
<?php
if (expr){
    sentencia;
}
?>
```

Si expr se evalúa como TRUE, PHP ejecutará la sentencia, y si se evalúa como FALSE - la ignorará o seguirá a la siguiente línea de código. Las sentencias if se pueden anidar indefinidamente dentro de otras sentencias if, lo cual proporciona una flexibilidad completa para ejecuciones condicionales en las diferentes partes de un programa.

#### 2.1.3.2. Else

A menudo se quiere ejecutar una sentencia si se cumple una cierta condición, y una sentencia distinta si la condición no se cumple.

Else extiende una sentencia if para ejecutar una sentencia en caso de que la expresión en la sentencia if se evalúe como FALSE. Por ejemplo, el siguiente código mostraría a es mayor que b si \$a fuera mayor que \$b, y a NO es mayor que b en cualquier otro caso:

```
<?php
if ($a > $b) { print "a es mayor que b"; }
else { print "a NO es mayor que b"; }
?>
```

---

<sup>17</sup> De la Cruz Villar, Joel. Op. Cit. p. 60.

La sentencia else se ejecuta solamente si la expresión if se evalúa como FALSE, y si hubiera alguna expresión elseif - sólo si se evaluaron también a FALSE

### 2.1.3.3. Elseif

Elseif, como su nombre sugiere, es una combinación de if y else. Como else, extiende una sentencia if para ejecutar una sentencia diferente en caso de que la expresión if original se evalúa como FALSE. No obstante, a diferencia de else, ejecutará esa expresión alternativa solamente si la expresión condicional elseif se evalúa como TRUE<sup>18</sup>.

Por ejemplo, el siguiente código mostraría a es mayor que b, a es igual a b o a es menor que b:

```
<?php
    if ($a > $b) {
        print "a es mayor que b";
    }
    elseif ($a == $b) {
        print "a es igual que b";
    }
    else {
        print "a es mayor que b";
    }
?>
```

Puede haber varios elseifs dentro de la misma sentencia if. La primera expresión elseif (si hay alguna) que se evalúe como TRUE se ejecutaría. En PHP, también se puede escribir 'else if' (con dos palabras) y el comportamiento sería idéntico al de un 'elseif' (una sola palabra).

La sentencia elseif se ejecuta sólo si la expresión if precedente y cualquier expresión elseif precedente se evalúan como FALSE, y la expresión elseif actual se evalúa como TRUE.

### 2.1.3.4. Sintaxis alternativa de estructuras de control

PHP ofrece una sintaxis alternativa para alguna de sus estructuras de control; a saber, if, while, for, y switch. En cada caso, la forma básica de la sintaxis alternativa es cambiar abrir-llave por dos puntos (:) y cerrar-llave

<sup>18</sup> De la Cruz Villar, Joel. Op. Cit. p. 63.

por endif;, endwhile;, endfor;, or endswitch;, respectivamente. Ejemplo:

```
<?php
    if ($a==5):
?>
    A es igual a 5
<?php endif;
?>
```

En el ejemplo de arriba, el bloque HTML "A es igual 5" se anida dentro de una sentencia if escrita en la sintaxis alternativa. El bloque HTML se mostraría solamente si \$a fuera igual a 5.

La sintaxis alternativa se aplica a else y también a elseif. La siguiente es una estructura if con elseif y else en el formato alternativo:

```
<?php
    if ($a == 5):
        print "a es igual a 5";
        print "...";
    elseif ($a == 6):
        print "a es igual a 6";
        print "!!!";
    else:
        print "a no es ni 5 ni 6";
    endif;
?>
```

#### 2.1.3.5. While

Los bucles while son los tipos de bucle más simples en PHP. Se comportan como su contrapartida en C. La forma básica de una sentencia while es:

```
while (expr) sentencia
```

El significado de una sentencia while es simple. Le dice a PHP que ejecute la(s) sentencia(s) anidada(s) repetidamente, mientras la expresión while se evalúe como TRUE. El valor de la expresión es comprobado cada vez al principio del bucle, así que incluso si este valor cambia durante la ejecución de la(s) sentencia(s) anidada(s), la ejecución no parará hasta el fin de la iteración (cada vez que PHP ejecuta las sentencias en el bucle es una iteración). A veces, si la expresión while se evalúa como FALSE desde el principio de todo, la(s) sentencia(s) anidada(s) no se ejecutarán ni siquiera una vez. Como con la sentencia if, se pueden agrupar múltiples sentencias

dentro del mismo bucle `while` encerrando un grupo de sentencias con llaves, o usando la sintaxis alternativa<sup>19</sup>:

`while (expr): sentencia ... endwhile;`

Los ejemplos son idénticos, y ambos imprimen números del 1 al 10:

```
<?php
/* ejemplo 1 */
$i = 1;
while ($i <= 10) {
    print $i++; /* el valor impreso será; a
    $i antes del incremento
    (post-incremento) */
}
/* ejemplo 2 */
$i = 1;
while ($i <= 10):
    print $i;
    $i++;
endwhile;
?>
```

#### 2.1.3.6. Do...While

Los bucles `do..while` son muy similares a los bucles `while`, excepto que las condiciones se comprueban al final de cada iteración en vez de al principio. La principal diferencia frente a los bucles regulares `while` es que se garantiza la ejecución de la primera iteración de un bucle `do..while` (la condición se comprueba sólo al final de la iteración), mientras que puede no ser necesariamente ejecutada con un bucle `while` regular (la condición se comprueba al principio de cada iteración, si esta se evalúa como `FALSE` desde el principio la ejecución del bucle finalizará inmediatamente)<sup>20</sup>. Hay una sola sintaxis para los bucles `do..while`:

```
<?php
    $i = 0;
    do {
        print $i;
    } while ($i>0);
?>
```

<sup>19</sup> De la Cruz Villar, Joel. *PHP y MySQL*. Lima: Megabyte, 2006, p. 65.

<sup>20</sup> De la Cruz Villar, Joel. Op. Cit. p. 66 - 68.

El bucle de arriba se ejecutaría exactamente una sola vez, después de la primera iteración, cuando la condición se comprueba, se evalúa como FALSE (\$i no es más grande que 0) y la ejecución del bucle finaliza.

Para permitir parar la ejecución en medio de los bloques de código, encapsulándolos con do..while(0), se usa la sentencia break.

El siguiente fragmento de código demuestra esto:

```
<?php
do {
    if ($i < 5) {
        print "i no es lo suficientemente grande";
        break;
    }
    $i *= $factor;
    if ($i < $minimum_limit) {
        break;
    }
    print "i es correcto";
} /* procesa i */ while(0);
?>
```

#### 2.1.3.7. For

El bucle For no es estrictamente necesario, cualquier bucle For puede ser sustituido fácilmente por otro WHILE. El bucle For resulta muy útil para ejecutar un bloque de código a condición de que una variable se encuentre entre un valor mínimo y otro máximo.

El bucle For también se puede romper mediante la sentencia BREAK.

```
<?php
for ($num = 1; num <=5; $num++){
    echo $num;
    if ($num == 3){
        echo "Aquí nos salimos \n";
        break;
    }
}
?>
```

#### 2.1.3.8. Switch

La estructura switch permite sustituir una serie de "if" en la misma expresión y, actuar independientemente del valor de ésta:

```
<?php
    switch ($i) {
        case 0: echo "\$i vale 0";
        break;
        case 1: echo "\$i vale 1";
        break; }
?>
```

Aquí se ha introducido "break" que permite salir de un bloque en el caso de que se realice una condición establecida.

## 2.1.4. FUNCIONES

Una función consiste en un conjunto de rutinas y acciones que a lo largo de el script PHP van a ser ejecutadas multitud de veces agrupados en una FUNCIÓN y desde cualquier punto del script PHP puede ser llamada y ejecutada. A su vez, esta función puede recibir parámetros externos de los cuales dependa el resultado de una función.

### 2.1.4.1. Sintaxis de una función

Las funciones deben ser colocadas siempre antes de realizar la llamada a la función. La sintaxis de una función es la siguiente:

```
function nombre(parámetros){
    instrucciones de la función
}
```

Para llamar a la función sería de la siguiente forma: nombre(parámetros)

```
<?php

function sumar($sumando1,$sumando2){
    $ suma=$sumando1+$sumando2
    echo $sumando1."+".$sumando2."=".$suma;
}
sumar(5,6)
?>
```

Cabe destacar es que las variables que declaremos dentro de la función solo existirán o tendrán dicho valor dentro de la función. Existen casos en los cuales no sabemos el número de parámetros que le pasaremos a la función y en estos casos debemos usar las funciones creadas al efecto como son:

*func\_num\_args()*: número de parámetros que se le han pasado a la función.

*func\_get\_args()*: devuelve un elemento de los que forman la lista de argumentos.

#### 2.1.4.2. Instrucción Return

Cuando se invoca una función, la ejecución del programa pasa a ejecutar las líneas de código que contenga la función, y una vez terminado, el programa continúa su ejecución desde el punto en que fue llamada la función. Existe una manera de terminar la ejecución de la función aunque aún haya código por ejecutar, mediante el uso de la instrucción return que termina la ejecución del código de una función y devuelve un valor.

Se puede tener varios return en una función, cuantos más return se tenga menos reutilizable será la función.

```
<?php
function mayor ($x, $y){
    $msg = "";
    if ($x > $y) {
        $msg = $x." es mayor que ".$y;
    }
    else {
        $msg = $y." es mayor que ".$x;}
    return $msg;}
?>
```

Con la instrucción return puede devolverse cualquier tipo de valor, incluyendo tablas y objetos. PHP solo permite a las funciones devolver un valor, y si se quiere que una función devuelva varios se tiene que utilizar una tabla (array).

#### 2.1.4.3. Parámetros de las funciones

Existen dos formas de pasar los parámetros a una función, por valor o por referencia. Cuando se pasa una variable por valor a una función, en nada modificará el contenido de la variable. Mientras que si se hace por referencia, cualquier cambio en la función sobre la variable lo hará para siempre. En PHP, las variables se pasan por valor. Para hacerlo por referencia se debe anteponer un ampersand (&) a la variable.

```

<?php
function suma ($x, $y){
    $x = $x + 1;
    return $x+$y; }
$a = 1; $b = 2;
//parámetros por valor
echo suma ($a, $b); // imprimirá 4
echo $a; // imprimirá 1
//parámetros por referencia
echo suma (&$a, $b); // imprimirá 4
echo $a; // imprimirá 2
?>

```

Si se quiere que un parámetro de una función se pase siempre por referencia se debe anteponer un ampersand (&) al nombre del parámetro en la definición de la función.

En PHP se puede definir valores por defecto para los parámetros de una función. Estos valores tienen que ser una expresión constante, y no una variable o miembro de una clase. Además cuando se usa parámetros por defecto, éstos deben estar a la derecha de cualquier parámetro sin valor por defecto, de otra forma PHP devolverá un error.

```

<?php
function suma ($x=1, $y){
    $x = $x + 1;
    return $x+$y;
}
?>

```

Si se ejecuta esta función daría error, ya que se ha dado a \$x el valor 1 por defecto y se ha colocado a la izquierda de un parámetro que no tiene valor por defecto. La forma correcta es:

```

<?php
function suma ($y, $x=1){
    $x = $x + 1;
    return $x+$y;
}

```

#### 2.1.4.4. Variables en las funciones

Las variables estáticas se definen dentro de una función, la primera vez que es llamada dicha función la variable se inicializa, guardando su valor para posteriores llamadas.



```
<?php
function contador (){
    static $count = 0;
    $count = $count + 1;
    return $count;}
echo contador()."<BR>"; // imprimirá 1
echo contador()."<BR>"; // imprimirá 2
echo contador()."<BR>"; // imprimirá 3
?>
```

Las variables globales, no se pueden declarar dentro de una función, se llama a una variable que ya ha sido declarada, tomando el valor que tenga en ese momento, pudiendo ser modificado en la función.

```
<?php
var $a = 1;
function ver_a(){
    global $a;
    echo $a."<BR>"; // imprimirá el valor de $a
    $a += 1; // sumamos 1 a $a }
    echo ver_a(); // imprimirá 1
    echo ver_a(); // imprimirá 2
    $a = 7;
    echo ver_a(); // imprimirá 7
    echo ver_a(); // imprimirá 8
?>
```

#### 2.1.4.5. Funciones variables

PHP soporta el concepto de funciones variables, si una variable tiene unos paréntesis añadidos al final, PHP buscará una función con el mismo nombre que el contenido de la variable, e intentará ejecutarla.

```
<?php
function imprime($texto) {
    echo $texto . "\n";}
function imprimeNegrilla($texto){
    echo "<B>$texto</B>\n"; }
$MiFunc = "imprime";
$MiFunc("Hola"); //imprimirá Hola
$MiFunc = "imprimeNegrilla";
$MiFunc("Hola"); //imprimirá Hola
?>
```

### 2.1.2. APLICACIONES DE SEGURIDAD EN PHP

La seguridad en PHP es algo que no se debe dejar de lado, hay diferentes formas de asegurar el código, pero la principal siempre es: comprobar todo lo que pueda ser enviado por el usuario hacia el sitio Web.

### 2.2.1. SESIONES EN PHP

Uno de los aspectos más interesantes en el desarrollo Web con PHP es el uso de sesiones para controlar entre otras cosas la forma en la cual acceden los usuarios de una aplicación Web a los recursos de la misma, esto mediante la asignación de un identificador único para cada usuario. Con ello se puede conseguir hacer un seguimiento de cada usuario a lo largo de su visita a una página Web, desde que entra hasta que sale.

Se utiliza sesiones para almacenar una serie de variables en el servidor relacionándolas con un usuario concreto para que estén disponibles en diferentes páginas de un sitio Web.

Una vez almacenada una variable de sesión, se podrá leerla desde todas las páginas de la Web mientras la sesión no se destruya<sup>21</sup>.

#### 2.2.1.1. Guardar una variable

El soporte para sesiones de PHP permite inicializar fácilmente una sesión utilizando `session_start` y gracias al vector superglobal `$_SESSION`, se podrá trabajar con sesiones como si de cualquier otro vector se tratara:

```
<?php
// Inicializamos sesión
session_start();
// Guardamos una variable
$_SESSION['nombre'] = 'Fernando Díaz';
?>
```

#### 2.2.1.2. Leer una variable guardada

Si se ha guardado una sesión, se podrá acceder a sus variables desde todas las páginas PHP de la Web cuando sean visitadas por el mismo

---

<sup>21</sup> Sesiones en Php. [en línea].<<http://www.programacionweb.net/articulos/articulo/?num=377>>

usuario. Inicializando la sesión y con el mismo vector \$\_SESSION, para ver si existe una variable de sesión determinada, se usará isset como con cualquier otra variable:

```
<?php
// Inicializamos sesión
session_start();
// Comprobamos si existe la variable
if ( isset ( $_SESSION['nombre'] ) ) {
// Si existe
echo $_SESSION['nombre'];
} else {
// Si no existe
echo 'desconocido';
}
?>
```

#### 2.2.1.3. Borrar una variable

Se puede borrar una variable de sesión usando la función unset, pero además se puede eliminar la sesión con session\_destroy:

```
<?php
// Inicializamos sesión
session_start();
// Borramos la variable
unset ( $_SESSION['nombre'] ) {
// Borramos toda la sesión
session_destroy();
}
?>
```

#### 2.2.1.4. Funciones relacionadas con sesiones

Aquí algunas de las funciones asociadas a las sesiones más usadas<sup>22</sup>.

- **session\_register()**

Para registrar variables. Acepta como argumento una cadena que identifique el nombre de una variable, o una matriz que contenga más identificadores de variables.

- **session\_unregister()**

Para eliminar alguna variable registrada. Se le pasa como parámetro el nombre de la variable.

---

<sup>22</sup> Sesiones en Php. [en línea].< <http://www.programacionweb.net/articulos/articulo/?num=573>>

- **string session\_id ()**  
Lee o cambia el identificador de sesión actual.
- **session\_name()**  
Devuelve el nombre de la sesión.
- **void session\_unset ( )**  
Elimina todas las variables de la sesión.
- **bool session\_is\_registered()**  
Comprueba si una variable está registrada en la sesión.
- **string session\_save\_path()**  
Lee o cambia la ruta donde se guardan los datos de la sesión actual.  
Hay que configurar el save\_path para que no de error en php.ini.
- **string session\_module\_name ( )**  
Lee y/o cambia el módulo de la sesión actual.
- **bool session\_set\_save\_handler()**  
Establece unas funciones para el almacenamiento de los datos de la sesión a nivel de usuario.
- **array session\_get\_cookie\_params()**  
Obtiene los parámetros de la cookie de la sesión.
- **void session\_set\_cookie\_params()**  
Cambia los parámetros de la cookie de la sesión.
- **void session\_write\_close()**  
Escribe los datos de la sesión y la finaliza.
- **bool session\_regenerate\_id()**  
Actualizar la id de sesión actual con una recién generada.
- **string session\_encode()**  
Devuelve una cadena con el contenido de la sesión actual en su interior.

## 2.3. INTRODUCCIÓN A JAVASCRIPT

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas Web dinámicas.

Una página Web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario.

Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. Los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios. A pesar de su nombre, JavaScript no guarda ninguna relación directa con el lenguaje de programación Java.

### **2.3.1. BREVE HISTORIA**

A principios de los años 90, la mayoría de usuarios que se conectaban a Internet lo hacían con módems a una velocidad máxima de 28.8 kbps. En esa época, empezaban a desarrollarse las primeras aplicaciones Web y por tanto, las páginas Web comenzaban a incluir formularios complejos.

Con unas aplicaciones Web cada vez más complejas y una velocidad de navegación tan lenta, surgió la necesidad de un lenguaje de programación que se ejecutará en el navegador del usuario. De esta forma, si el usuario no rellenaba correctamente un formulario, no se le hacía esperar mucho tiempo hasta que el servidor volviera a mostrar el formulario indicando los errores existentes.

Brendan Eich, un programador que trabajaba en Netscape, pensó que podría solucionar este problema adaptando otras tecnologías existentes (como *ScriptEase*) al navegador Netscape Navigator 2.0, que iba a lanzarse en 1995. Inicialmente, Eich denominó a su lenguaje *LiveScript*.

Posteriormente, Netscape firmó una alianza con Sun Microsystems para el desarrollo del nuevo lenguaje de programación. Además, justo antes del lanzamiento Netscape decidió cambiar el nombre por el de JavaScript. La razón del cambio de nombre fue exclusivamente por marketing, ya que Java era la palabra de moda en el mundo informático y de Internet de la época.

La primera versión de JavaScript fue un completo éxito y Netscape Navigator 3.0 ya incorporaba la siguiente versión del lenguaje, la versión 1.1.

Al mismo tiempo, Microsoft lanzó JScript con su navegador Internet Explorer 3.

JScript era una copia de JavaScript al que le cambiaron el nombre para evitar problemas legales. Para evitar una guerra de tecnologías, Netscape decidió que lo mejor sería estandarizar el lenguaje JavaScript. De esta forma, en 1997 se envió la especificación JavaScript 1.1 al organismo ECMA (European Computer Manufacturers Association ).

ECMA creó el comité TC39 con el objetivo de estandarizar de un lenguaje de script multiplataforma e independiente de cualquier empresa.

El primer estándar que creó el comité TC39 se denominó ECMA-262, en el que se definió por primera vez el lenguaje ECMAScript. Por este motivo, algunos programadores prefieren la denominación *ECMAScript* para referirse al lenguaje JavaScript. JavaScript no es más que la implementación que realizó la empresa Netscape del estándar ECMAScript.

La organización internacional para la estandarización (ISO) adoptó el estándar ECMA-262 a través de su comisión IEC, dando lugar al estándar ISO/IEC-16262.

ECMA también ha definido varios estándares relacionados con ECMAScript, como el estándar ECMA-357, que define una extensión conocida como E4X y que permite la integración de JavaScript y XML<sup>23</sup>.

### 2.3.2. SINTAXIS

La sintaxis de un lenguaje de programación se define como el conjunto de reglas que deben seguirse al escribir el código fuente de los programas para considerarse como correctos para ese lenguaje de programación.

La sintaxis de JavaScript es muy similar a la de otros lenguajes de programación como Java y C. Las normas básicas que definen la sintaxis de JavaScript son las siguientes:

- No se tienen en cuenta los espacios en blanco y las nuevas líneas: como sucede con XHTML<sup>24</sup>, el intérprete de JavaScript ignora cualquier espacio en blanco sobrante, por lo que el código se puede ordenar de forma adecuada para

---

<sup>23</sup> JavaScript. [en línea]. < <http://www.librosweb.es/javascript>>

<sup>24</sup> Lenguaje de Marcado de Hipertexto Extensible, con un etiquetado más estricto que HTML, que permite una correcta interpretación de la información, independientemente del dispositivo con que se accede a ella.

entenderlo mejor (tabulando las líneas, añadiendo espacios, creando nuevas líneas, etc.)

- Se distinguen las mayúsculas y minúsculas: al igual que sucede con la sintaxis de las etiquetas y elementos XHTML. Sin embargo, si en una página XHTML se utilizan indistintamente mayúsculas y minúsculas, la página se visualiza correctamente, siendo el único problema la no validación de la página. En cambio, si en JavaScript se intercambian mayúsculas y minúsculas el script no funciona.
- No se define el tipo de las variables: al crear una variable, no es necesario indicar el tipo de dato que almacenará. De esta forma, una misma variable puede almacenar diferentes tipos de datos durante la ejecución del script.
- No es necesario terminar cada sentencia con el carácter de punto y coma (;): en la mayoría de lenguajes de programación, es obligatorio terminar cada sentencia con el carácter; .Aunque JavaScript no obliga a hacerlo, es conveniente seguir la tradición de terminar cada sentencia con el carácter del punto y coma (;).
- Se pueden incluir comentarios: los comentarios se utilizan para añadir información en el código fuente del programa. Aunque el contenido de los comentarios no se visualiza por pantalla, sí que se envía al navegador del usuario junto con el resto del script, por lo que es necesario extremar las precauciones sobre la información incluida en los comentarios.
- JavaScript define dos tipos de comentarios: los de una sola línea y los que ocupan varias líneas.

Ejemplo de comentario de una sola línea:

```
// A continuación se muestra un mensaje  
alert("mensaje de prueba");
```

Ejemplo de comentario de varias líneas:

```
/* Los comentarios de varias líneas son muy útiles cuando se necesita  
incluir bastante información en los comentarios */  
alert("mensaje de prueba");
```

### **2.3.3. JAVASCRIPT Y OTROS NAVEGADORES**

Los navegadores más modernos disponibles actualmente incluyen soporte de JavaScript hasta la versión correspondiente a la tercera edición del estándar

ECMA-262. La mayor diferencia reside en el dialecto utilizado, ya que mientras Internet Explorer utiliza JScript, el resto de navegadores (Firefox, Opera, Safari, Konqueror) utilizan JavaScript.

#### 2.3.4. JAVASCRIPT EN OTROS ENTORNOS

La inigualable popularidad de JavaScript como lenguaje de programación de aplicaciones Web se ha extendido a otras aplicaciones y otros entornos no relacionados con la Web.

Herramientas como Adobe Acrobat permiten incluir código JavaScript en archivos PDF. Otras herramientas de Adobe como Flash y Flex utilizan ActionScript, un dialecto del mismo estándar de JavaScript.

PhotoShop permite realizar pequeños scripts mediante JavaScript y la versión 6 de Java incluye un nuevo paquete (denominado javax.script) que permite integrar ambos lenguajes. Aplicaciones como Yahoo Widgets y el Dashboard de Apple utilizan JavaScript para programar sus widgets<sup>25</sup>.

#### 2.3.5. VARIABLES

Una variable es un elemento que se emplea para almacenar y hacer referencia a otro valor. Las variables en JavaScript se crean mediante la palabra reservada `var`.

```
var numero_1 = 3;  
var numero_2 = 1;  
var resultado = numero_1 + numero_2;
```

La palabra reservada `var` solamente se debe indicar al definir por primera vez la variable, lo que se denomina declarar una variable.

Cuando se utilizan las variables en el resto de instrucciones del script, solamente es necesario indicar su nombre. El nombre de una variable también se conoce como identificador y debe cumplir las siguientes normas:

- Sólo puede estar formado por letras, números y los símbolos \$ (dólar) y \_ (guión bajo).

---

<sup>25</sup> Un widget es una aplicación interactiva de propósito único que muestra y/o actualiza datos locales o en la Web y que está paquetizada de manera que permita una simple descarga e instalación en la máquina del usuario o en un dispositivo móvil.



- El primer carácter no puede ser un número. Por tanto, las siguientes variables tienen nombres correctos:

```
var $numero1;  
var _$letra;  
var $$$otroNumero;  
var $_a__$4;
```

Sin embargo, las siguientes variables tienen identificadores incorrectos:

```
var 1numero; // Empieza por un número  
var numero;1_123; // Contiene un carácter ";"
```

### 2.3.6. TIPOS DE VARIABLES

Aunque todas las variables de JavaScript se crean de la misma forma (mediante la palabra reservada `var`), la forma en la que se les asigna un valor depende del tipo de valor que se quiere almacenar (números, textos, etc.)

#### 2.3.6.1. Numéricas

Se utilizan para almacenar valores numéricos enteros (llamados *integer* en inglés) o decimales (llamados *float* en inglés). En este caso, el valor se asigna indicando directamente el número entero o decimal. Los números decimales utilizan el carácter `.` (punto) en vez de `,` (coma) para separar la parte entera y la parte decimal:

```
var iva = 16; // variable tipo entero  
var total = 234.65; // variable tipo decimal
```

#### 2.3.6.2. Cadenas de texto

Se utilizan para almacenar caracteres, palabras y/o frases de texto. Para asignar el valor a la variable, se encierra el valor entre comillas dobles o simples, para delimitar su comienzo y su final:

```
var mensaje = "Bienvenido a nuestro sitio web";  
var nombreProducto = 'Producto ABC';  
var letraSeleccionada = 'c';
```

En ocasiones, el texto que se almacena en las variables no es tan sencillo. Si por ejemplo el propio texto contiene comillas simples o

dobles, la estrategia que se sigue es la de encerrar el texto con las comillas (simples o dobles) que no utilice el texto:

```
/* El contenido de texto1 tiene comillas simples, por lo que se
encierra con comillas dobles */
var texto1 = "Una frase con 'comillas simples' dentro";
/* El contenido de texto2 tiene comillas dobles, por lo que se
encierra con comillas simples */
var texto2 = 'Una frase con "comillas dobles" dentro';
```

A veces las cadenas de texto contienen tanto comillas simples como dobles. Además, existen otros caracteres que son difíciles de incluir en una variable de texto (tabulador, ENTER, etc.).

Para resolver estos problemas, JavaScript define un mecanismo para incluir de forma sencilla caracteres especiales y problemáticos dentro de una cadena de texto. El mecanismo consiste en sustituir el carácter problemático por una combinación simple de caracteres. Ejemplo:

Si se quiere incluir...	Se debe incluir...
Una nueva línea	\n
Un tabulador	\t
Una comilla simple	\'
Una comilla doble	\"
Una barra inclinada	\\

*Tabla 2.8. Tabla de Conversión*

De esta forma, el ejemplo anterior que contenía comillas simples y dobles dentro del texto se puede rehacer de la siguiente forma:

```
var texto1 = 'Una frase con \'comillas simples\' dentro';
var texto2 = "Una frase con \"comillas dobles\" dentro";
```

Este mecanismo de JavaScript se denomina "mecanismo de escape" de los caracteres problemáticos, y es habitual referirse a que los caracteres han sido "escapados".

### 2.3.6.3. Arrays

Un array es una colección de variables, que pueden ser todas del mismo tipo o cada una de un tipo diferente. Ejemplo:

Si una aplicación necesita manejar los días de la semana.

```
var dias = ["Lunes", "Martes", "Miércoles", "Jueves",
            "Viernes", "Sábado", "Domingo"];
```

Para definir un array, se utilizan los caracteres [ y ] para delimitar su comienzo y su final y se utiliza el carácter , (coma) para separar sus elementos:

```
var nombre_array = [valor1, valor2, ..., valorN];
```

Una vez definido un array se accede a cada elemento indicando su posición dentro del array. Las posiciones de los elementos empiezan a contarse en el 0 y no en el 1:

```
var diaSeleccionado = dias[0]; // diaSeleccionado = "Lunes"
var otroDia = dias[5]; // otroDia = "Sábado"
```

#### 2.3.6.4. Booleanos

Las variables de tipo boolean o booleano también se conocen con el nombre de variables de tipo lógico. Una variable de tipo boolean almacena un tipo de valor que solamente puede tomar dos valores: true (verdadero) o false (falso). No se utiliza para almacenar números y tampoco permite cadenas de texto. Los únicos valores que pueden almacenar estas variables son true y false, por lo que no pueden utilizarse los valores verdadero y falso.

```
var clienteRegistrado = false;
var ivaIncluido = true;
```

### 2.3.7. OPERADORES

Los operadores permiten manipular el valor de las variables, realizar operaciones matemáticas con sus valores y comparar diferentes variables. Los operadores permiten a los programas realizar cálculos complejos y tomar decisiones lógicas en función de comparaciones y otros tipos de condiciones.

#### 2.3.7.1. Asignación

Este operador se utiliza para guardar un valor específico en una variable. El símbolo utilizado es (=).

```

var numero1 = 3;
var numero2 = 4;
/* Error, la asignación siempre se realiza a una variable, por lo
que en la izquierda no se puede indicar un número */
5 = numero1;
// Ahora, la variable numero1 vale 5
numero1 = 5;
// Ahora, la variable numero1 vale 4
numero1 = numero2;

```

### 2.3.7.2. Incremento y decremento

Estos dos operadores solamente son válidos para las variables numéricas y se utilizan para incrementar o decrementar en una unidad el valor de una variable. Ejemplo:

```

var numero = 5;
++numero;
alert(numero); // numero = 6

```

El operador de incremento se indica mediante el prefijo ++ en el nombre de la variable. El resultado es que el valor de esa variable se incrementa en una unidad.

```

var numero = 5;
numero = numero + 1;
alert(numero); // numero = 6

```

El operador decremento (indicado como un prefijo - en el nombre de la variable) se utiliza para decrementar el valor de la variable:

```

var numero = 5;
--numero;
alert(numero); // numero = 4
Es equivalente a:
var numero = 5;
numero = numero - 1;
alert(numero); // numero = 4

```

Estos operadores también se los utiliza como sufijo. Ejemplo:

```

var numero = 5;
numero++;
alert(numero); // numero = 6

```

Si el operador ++ se indica como prefijo del identificador de la variable, su valor se incrementa antes de realizar cualquier otra operación.

Si el operador ++ se indica como sufijo del identificador de la variable, su valor se incrementa después de ejecutar la sentencia en la que aparece.

### 2.3.7.3. Lógicos

Los operadores lógicos son imprescindibles para realizar aplicaciones complejas, ya que se utilizan para tomar decisiones sobre las instrucciones que debería ejecutar el programa en función de ciertas condiciones.

El resultado de cualquier operación que utilice operadores lógicos siempre es un valor lógico o booleano.

### 2.3.7.4. Negación

Uno de los operadores lógicos más utilizados es el de la negación. Se utiliza para obtener el valor contrario al valor de la variable:

```
var visible = true;
alert(!visible); // Muestra "false" y no "true"
```

La negación lógica se obtiene prefijando el símbolo ! al identificador de la variable. El funcionamiento se resume en la siguiente tabla:

variable	!variable
true	false
false	true

Tabla 2.9. Operadores Lógicos: Negación

### 2.3.7.5. AND

La operación lógica AND obtiene su resultado combinando dos valores booleanos. El operador se indica mediante el símbolo && y su resultado solamente es true si los dos operandos son true:

Variable1	Variable2	Variable1&&variable2
true	true	true
true	false	false
false	true	false
false	false	false

Tabla 2.10. Operadores Lógicos: AND

```

var valor1 = true;
var valor2 = false;
resultado = valor1 && valor2; // resultado = false
valor1 = true;
valor2 = true;
resultado = valor1 && valor2; // resultado = true

```

### 2.3.7.6. OR

La operación lógica OR también combina dos valores booleanos. El operador se indica mediante el símbolo || y su resultado es true si alguno de los dos operandos es true:

Variable1	Variable2	Variable    variable2
true	true	true
true	false	true
false	true	true
false	false	false

Tabla 2.11. Operadores Lógicos: OR

```

var valor1 = true;
var valor2 = false;
resultado = valor1 || valor2; // resultado = true
valor1 = false;
valor2 = false;
resultado = valor1 || valor2; // resultado = false

```

### 2.3.7.7. Matemáticos

JavaScript permite realizar manipulaciones matemáticas sobre el valor de las variables numéricas. Los operadores definidos son: suma (+), resta (-), multiplicación (\*) y división (/).

```

var numero1 = 10;
var numero2 = 5;
resultado = numero1 / numero2; // resultado = 2
resultado = 3 + numero1; // resultado = 13
resultado = numero2 - 4; // resultado = 1
resultado = numero1 * numero2; // resultado = 50

```

### 2.3.7.8. Relacionales

Los operadores relacionales definidos por JavaScript son idénticos a los que definen las matemáticas: mayor que (>), menor que (<), mayor o igual (>=), menor o igual (<=), igual que (==) y distinto de (!=). Los operadores

que relacionan variables son imprescindibles para realizar cualquier aplicación compleja. El resultado de todos estos operadores siempre es un valor booleano:

```
var numero1 = 3;
var numero2 = 5;
resultado = numero1 > numero2; // resultado = false
resultado = numero1 < numero2; // resultado = true
numero1 = 5;
numero2 = 5;
resultado = numero1 >= numero2; // resultado = true
resultado = numero1 <= numero2; // resultado = true
resultado = numero1 == numero2; // resultado = true
resultado = numero1 != numero2; // resultado = false
```

Los operadores relacionales también se pueden utilizar con variables de tipo cadena de texto:

```
var texto1 = "hola";
var texto2 = "hola";
var texto3 = "adios";
resultado = texto1 == texto3; // resultado = false
resultado = texto1 != texto2; // resultado = false
resultado = texto3 >= texto2; // resultado = false
```

Cuando se utilizan cadenas de texto, los operadores "mayor que" (>) y "menor que" (<) siguen un razonamiento no intuitivo: se compara letra a letra comenzando desde la izquierda hasta que se encuentre una diferencia entre las dos cadenas de texto. Para determinar si una letra es mayor o menor que otra, las mayúsculas se consideran menores que las minúsculas y las primeras letras del alfabeto son menores que las últimas (a es menor que b, b es menor que c, A es menor que a, etc.)

### 2.3.8. ESTRUCTURAS DE CONTROL DE FLUJO

Los programas que se pueden realizar utilizando solamente variables y operadores son una simple sucesión lineal de instrucciones básicas.

Las estructuras de control de flujo que son instrucciones del tipo "si se cumple esta condición, hazlo; si no se cumple, haz esto otro". También existen instrucciones del tipo "repite esto mientras se cumpla esta condición"<sup>26</sup>.

---

<sup>26</sup> Estructuras de Control. [En línea]. <[http://www.librosweb.es/javascript/capitulo3/estructuras\\_de\\_control\\_de\\_flujo.html](http://www.librosweb.es/javascript/capitulo3/estructuras_de_control_de_flujo.html)>

Si se utilizan estructuras de control de flujo, los programas dejan de ser una sucesión lineal de instrucciones para convertirse en programas inteligentes que pueden tomar decisiones en función del valor de las variables.

#### 2.3.8.1. Estructura If

La estructura más utilizada en JavaScript y en la mayoría de lenguajes de programación es la estructura if. Se emplea para tomar decisiones en función de una condición. Su definición formal es:

```
if(condicion) {  
    ...  
}
```

Si la condición se cumple (es decir, si su valor es true) se ejecutan todas las instrucciones que se encuentran dentro de {...}.

Si la condición no se cumple (es decir, si su valor es false) no se ejecuta ninguna instrucción contenida en {...} y el programa continúa ejecutando el resto de instrucciones del script.

```
var mostrarMensaje = true;  
if(mostrarMensaje) {  
    alert("Hola Mundo");  
}  
El ejemplo se podría reescribir también como:  
var mostrarMensaje = true;  
if(mostrarMensaje == true) {  
    alert("Hola Mundo");  
}
```

#### 2.3.8.2. Estructura If...Else

Existe una variante de la estructura if llamada if...else. Su definición formal es la siguiente:

```
if(condicion) {  
    ...  
}  
else {  
    ...  
}
```

Si la condición se cumple (es decir, si su valor es true) se ejecutan todas las instrucciones que se encuentran dentro del if(). Si la



condición no se cumple (es decir, si su valor es false) se ejecutan todas las instrucciones contenidas en else { }.

```
var edad = 18;
if(edad >= 18) {
  alert("Eres mayor de edad");
}
else {
  alert("Todavía eres menor de edad");
}
```

### 2.3.8.3. Estructura For

La estructura for permite realizar repeticiones (también llamadas bucles) de una forma muy sencilla. No obstante, su definición formal no es tan sencilla como la de if():

```
for(inicializacion; condicion; actualizacion) {
  ...
}
```

La idea del funcionamiento de un bucle for es la siguiente: "mientras la condición indicada se siga cumpliendo, repite la ejecución de las instrucciones definidas dentro del for. Además, después de cada repetición, actualiza el valor de las variables que se utilizan en la condición". La "inicialización" es la zona en la que se establece los valores iniciales de las variables que controlan la repetición. La "condición" es el único elemento que decide si continua o se detiene la repetición.

La "actualización" es el nuevo valor que se asigna después de cada repetición a las variables que controlan la repetición<sup>27</sup>.

```
var mensaje = "Hola, estoy dentro de un bucle";
for(var i = 0; i < 5; i++) {
  alert(mensaje);
}
```

### 2.3.8.4. Estructura For...In

Una estructura de control derivada de for es la estructura for...in. Su definición formal adaptada a los arrays es:

---

<sup>27</sup> Estructuras de Control. [En línea].

<[http://www.librosweb.es/javascript/capitulo3/estructuras\\_de\\_control\\_de\\_flujo.html](http://www.librosweb.es/javascript/capitulo3/estructuras_de_control_de_flujo.html)>

```
for(indice in array) {
    ...
}
```

Si se quieren recorrer todos los elementos que forman un array, la estructura for...in es la forma más eficiente de hacerlo.

```
var dias = ["Lunes", "Martes", "Miércoles", "Jueves", "Viernes",
            "Sábado",
            "Domingo"];
for(i in dias) {
    alert(dias[i]);
}
```

La variable que se indica como índice es la que se puede utilizar dentro del bucle for...in para acceder a los elementos del array. En la primera repetición del bucle la variable i vale 0 y en la última vale 6. Esta estructura de control es la más adecuada para recorrer arrays (y objetos), ya que evita tener que indicar la inicialización y las condiciones del bucle for simple y funciona correctamente sea la longitud del array.

### 2.3.9. FUNCIONES Y PROPIEDADES BÁSICAS DE JAVASCRIPT

JavaScript incorpora una serie de herramientas y utilidades para el manejo de las variables. De esta forma, muchas de las operaciones básicas con las variables, se pueden realizar directamente con las utilidades que ofrece JavaScript <sup>28</sup>.

#### 2.3.9.1. Funciones para cadenas de texto

A continuación algunas de las funciones más útiles para el manejo de cadenas de texto:

- `length`, calcula la longitud de una cadena de texto (el número de caracteres que la forman).

```
var mensaje = "Hola Mundo";
var numeroLetras = mensaje.length; // numeroLetras = 10
```

- `+`, se emplea para concatenar varias cadenas de texto.

<sup>28</sup> Funciones y Propiedades. [En línea].

<[http://www.librosweb.es/javascript/capitulo3/funciones\\_y\\_propiedades\\_basicas\\_de\\_javascript.html](http://www.librosweb.es/javascript/capitulo3/funciones_y_propiedades_basicas_de_javascript.html)>

```
var mensaje1 = "Hola";
var mensaje2 = " Mundo";
var mensaje = mensaje1 + mensaje2;
// mensaje = "Hola Mundo"
```

- concat(), también se puede utilizar la función para concatenar varias cadenas de texto.

```
var mensaje1 = "Hola";
var mensaje2 = mensaje1.concat(" Mundo");
// mensaje2 = "Hola Mundo"
```

Las cadenas de texto también se pueden unir con variables numéricas:

```
var variable1 = "Hola ";
var variable2 = 3;
var mensaje = variable1 + variable2; // mensaje = "Hola 3"
```

Cuando se unen varias cadenas de texto es habitual olvidar añadir un espacio de separación entre las palabras:

```
var mensaje1 = "Hola";
var mensaje2 = "Mundo";
var mensaje = mensaje1 + mensaje2; // mensaje = "HolaMundo"
```

Los espacios en blanco se pueden añadir al final o al principio de las cadenas y también se pueden indicar forma explícita:

```
var mensaje1 = "Hola";
var mensaje2 = "Mundo";
var mensaje = mensaje1 + " " + mensaje2;
// mensaje = "Hola Mundo"
```

- toUpperCase(), transforma todos los caracteres de la cadena a sus correspondientes caracteres en mayúsculas:

```
var mensaje1 = "Hola";
var mensaje2 = mensaje1.toUpperCase(); // mensaje2 = "HOLA"
```

- toLowerCase(), transforma todos los caracteres de la cadena a sus correspondientes caracteres en minúsculas:

```
var mensaje1 = "HoIA";
var mensaje2 = mensaje1.toLowerCase(); // mensaje2 = "hola"
```

- `charAt(posicion)`, obtiene el carácter que se encuentra en la posición indicada:

```
var mensaje = "Hola";  
var letra = mensaje.charAt(0); // letra = H  
letra = mensaje.charAt(2); // letra = l
```

- `indexOf(caracter)`, calcula la posición en la que se encuentra el carácter indicado dentro de la cadena de texto. Si el carácter se incluye varias veces dentro de la cadena de texto, se devuelve su primera posición empezando a buscar desde la izquierda. Si la cadena no contiene el carácter, la función devuelve el valor -1:

```
var mensaje = "Hola";  
var posicion = mensaje.indexOf('a'); // posicion = 3  
posicion = mensaje.indexOf('b'); // posicion = -1
```

- `lastIndexOf(caracter)`, calcula la última posición en la que se encuentra el carácter indicado dentro de la cadena de texto. Si la cadena no contiene el carácter, la función devuelve el valor -1:

```
var mensaje = "Hola";  
var posicion = mensaje.lastIndexOf('a'); // posicion = 3  
posicion = mensaje.lastIndexOf('b'); // posicion = -1
```

La función `lastIndexOf()` comienza su búsqueda desde el final de la cadena hacia el principio, aunque la posición devuelta es la correcta empezando a contar desde el principio de la palabra.

- `substring(inicio, final)`, extrae una porción de una cadena de texto. El segundo parámetro es opcional. Si sólo se indica el parámetro inicio, la función devuelve la parte de la cadena original correspondiente desde esa posición hasta el final:

```
var mensaje = "Hola Mundo";  
var porcion = mensaje.substring(2); // porcion = "la Mundo"  
porcion = mensaje.substring(5); // porcion = "Mundo"  
porcion = mensaje.substring(7); // porcion = "ndo"
```

Si se indica un inicio negativo, se devuelve la misma cadena original:

```
var mensaje = "Hola Mundo";  
var porcion = mensaje.substring(-2);  
// porcion = "Hola Mundo"
```

Cuando se indica el inicio y el final, se devuelve la parte de la cadena original comprendida entre la posición inicial y la inmediatamente anterior a la posición final (es decir, la posición inicio está incluida y la posición final no):

```
var mensaje = "Hola Mundo";
var porcion = mensaje.substring(1, 8); // porcion = "ola Mun"
porcion = mensaje.substring(3, 4); // porcion = "a"
```

Si se indica un final más pequeño que el inicio, JavaScript los considera de forma inversa, ya que automáticamente asigna el valor más pequeño al inicio y el más grande al final:

```
var mensaje = "Hola Mundo";
var porcion = mensaje.substring(5, 0); // porcion = "Hola "
porcion = mensaje.substring(0, 5); // porcion = "Hola "
```

- `split(separador)`, convierte una cadena de texto en un array de cadenas de texto. La función parte la cadena de texto determinando sus trozos a partir del carácter separador indicado:

```
var mensaje = "Hola Mundo, soy una cadena de texto!";
var palabras = mensaje.split(" ");
//palabras=["Hola", "Mundo,", "soy", "una", "cadena", "de", "texto!"];
```

Con esta función se pueden extraer fácilmente las letras que forman una palabra:

```
var palabra = "Hola";
var letras = palabra.split(""); // letras = ["H", "o", "l", "a"]
```

### 2.3.9.2. Funciones para arrays

Algunas de las funciones más útiles para el manejo de arrays:

- `length`, calcula el número de elementos de un array.

```
var vocales = ["a", "e", "i", "o", "u"];
var numeroVocales = vocales.length; // numeroVocales = 5
```

- `concat()`, se emplea para concatenar los elementos de varios arrays.

```
var array1 = [1, 2, 3];
array2 = array1.concat(4, 5, 6); // array2 = [1, 2, 3, 4, 5, 6]
array3 = array1.concat([4, 5, 6]); // array3 = [1, 2, 3, 4, 5, 6]
```

- `join(separador)`, es la función contraria a `split()`. Une todos los elementos de un array para formar una cadena de texto. Para unir los elementos se utiliza el carácter separador indicado.

```
var array = ["hola", "mundo"];
var mensaje = array.join(""); // mensaje = "holamundo"
mensaje = array.join(" "); // mensaje = "hola mundo"
```

- `pop()`, elimina el último elemento del array y lo devuelve. El array original se modifica y su longitud disminuye en 1 elemento.

```
var array = [1, 2, 3];
var ultimo = array.pop();
// ahora array = [1, 2], ultimo = 3
```

- `push()`, añade un elemento al final el array. El array original se modifica y aumenta su longitud en 1 elemento. (También es posible añadir más de un elemento a la vez).

```
var array = [1, 2, 3];
array.push(4);
// ahora array = [1, 2, 3, 4]
```

- `shift()`, elimina el primer elemento del array y lo devuelve. El array original se ve modificado y su longitud disminuida en 1 elemento.

```
var array = [1, 2, 3];
var primero = array.shift();
// ahora array = [2, 3], primero = 1
```

- `unshift()`, añade un elemento al principio del array. El array original se modifica y aumenta su longitud en 1 elemento.

```
var array = [1, 2, 3];
array.unshift(0);
// ahora array = [0, 1, 2, 3]
```

- `reverse()`, modifica un array colocando sus elementos en el orden inverso a su posición original:

```
var array = [1, 2, 3];
array.reverse();
// ahora array = [3, 2, 1]
```

### 2.3.9.3. Funciones para números

A continuación algunas de las funciones y propiedades más útiles para el manejo de números.

- NaN, (del inglés, *"Not a Number"*) JavaScript emplea el valor NaN para indicar un valor numérico no definido (por ejemplo, la división 0/0).

```
var numero1 = 0;
var numero2 = 0;
alert(numero1/numero2); // se muestra el valor NaN
```

- isNaN(), permite proteger a la aplicación de posibles valores numéricos no definidos.

```
var numero1 = 0;
var numero2 = 0;
if(isNaN(numero1/numero2)) {
    alert("La división no está definida para los números indicados");
}
else {
    alert("La división es igual a => " + numero1/numero2);
}
```

- Infinity, hace referencia a un valor numérico infinito y positivo (también existe el valor – Infinity para los infinitos negativos).

```
var numero1 = 10;
var numero2 = 0;
alert(numero1/numero2); // se muestra el valor Infinity
```

- toFixed(dígitos), devuelve el número original con tantos decimales como los indicados por el parámetro dígitos y realiza los redondeos necesarios. Se trata de una función muy útil por ejemplo para mostrar precios.

```
var numero1 = 4564.34567;
numero1.toFixed(2); // 4564.35
numero1.toFixed(6); // 4564.345670
numero1.toFixed(); // 4564
```

## 2.4. LENGUAJE DE MODELAMIENTO UNIFICADO – UML

UML (Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. La notación del UML viene de un esfuerzo colaborativo de Grandy Booch, James Rumbaugh, Ivar Jacobson, Rebecca Wirfs-Brock. Jacobson ha escrito acerca de cómo capturar los requisitos de los sistemas a través de paquetes de transiciones llamados casos de uso, también desarrolló un método para el diseño de sistemas al que llamó Ingeniería de Software Orientada a Objetos (OOSE) enfocado en el análisis.

Booch, Rumbaugh y Jacobson, volcaron todo su trabajo y experiencia para la formación de la Corporación del Software Rational enfocada en la estandarización y regularización y refinamiento del UML. Es importante resaltar que los símbolos de UML incluyen aspectos de Booch y notaciones de OMT (Object Modeling Technique) así como también elementos de otras notaciones.

UML permite a las personas desarrollar diferentes tipos de diagramas visuales que representan varios aspectos de los sistemas<sup>29</sup>.

### 2.4.1. JERARQUÍA DE LOS DIAGRAMAS UML

#### 2.4.1.1. Diagrama de Estructura

Los Diagramas de Estructura enfatizan en los elementos que deben existir en el sistema modelado<sup>30</sup>:

- Diagrama de clases

Los diagramas de clases muestran las interacciones entre las clases del sistema.

- Diagrama componentes

El diagrama de componentes permite mostrar una vista física del modelo, como también los componentes de software y las relaciones que existen entre ellos.

---

<sup>29</sup> Romero Moreno, Gesvin. *UML con Rational Rose*. Lima: Megabyte. 2004. p. 17-18.

<sup>30</sup> UML. [En línea]. <[http://es.wikipedia.org/wiki/Lenguaje\\_Unificado\\_de\\_Modelado](http://es.wikipedia.org/wiki/Lenguaje_Unificado_de_Modelado)>



- Diagrama de objetos

Los diagramas de objetos modelan las instancias de elementos contenidos en los diagramas de clases. Un diagrama de objetos muestra un conjunto de objetos y sus relaciones en un momento concreto.

- Diagrama de estructura compuesta

Un diagrama de estructura compuesta muestra la estructura interna de una clase y las colaboraciones que esta estructura hace posible.

- Diagrama de despliegue

Los diagramas de despliegue muestran el diseño físico de la red y el lugar donde residirán los componentes.

- Diagrama de paquetes

Un diagrama de paquetes muestra cómo un sistema está dividido en agrupaciones lógicas mostrando las dependencias entre esas agrupaciones. Los diagramas de paquetes suministran una descomposición de la jerarquía lógica de un sistema.

#### **2.4.1.2. Diagrama de Comportamiento**

Los Diagramas de Comportamiento enfatizan en lo que debe suceder en el sistema modelado:

- Diagrama de actividades

Los diagramas de actividades ilustran el flujo de funcionalidad en un sistema. Estos diagramas definen dónde empieza y dónde acaba el flujo del negocio y así conocer qué actividades ocurren durante el flujo y en qué orden ocurren.

- Diagrama casos de uso

Los diagramas de Caso de Uso hacen que se muestren las interacciones entre los casos de uso y los actores. Los casos de uso representan: la funcionalidad del sistema y los requisitos del sistema desde la perspectiva del usuario

- Diagrama de estados

Los diagramas de estado proporcionan una manera para modelar los

varios estados que pueden existir en un objeto. Los diagramas de estados se usan para modelar la conducta dinámica de un sistema.

#### **2.4.1.3. Diagramas de Interacción**

Los Diagramas de Interacción son un subtipo de diagramas de comportamiento, que enfatiza sobre el flujo de control y de datos entre los elementos del sistema modelado:

- Diagrama de secuencia

Los diagramas de secuencia se usan para mostrar el flujo de funcionalidad a través de un caso de uso.

- Diagrama comunicación

Un diagrama de comunicación modela las interacciones entre objetos o partes en términos de mensajes en secuencia. Los diagramas de comunicación representan una combinación de información tomada desde el diagrama de clases, secuencia, y diagrama de casos de uso describiendo tanto la estructura estática como el comportamiento dinámico de un sistema.

- Diagrama global de interacciones

Muestra una cierta vista sobre los aspectos dinámicos de los sistemas modelados. Aunque un diagrama global de interacciones es una representación gráfica de una interacción, éste se distingue fuertemente de los diagramas de secuencia y de comunicación. Algunos elementos gráficos del diagrama global de interacciones están tomados del diagrama de actividades.

## CAPÍTULO 3

### 3. ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE

#### 3.1. REQUERIMIENTOS FUNCIONALES

Los requisitos funcionales del sistema se desarrollarán en base a los requerimientos definidos por las funcionarias encargadas del Centro de Documentación y Difusión del Consejo Nacional de las Mujeres que serian en esencia para el eficiente funcionamiento del sistema a desarrollarse e implementarse.

Los requerimientos funcionales a implementar en el sistema son los siguientes:

- Crear un módulo para poder llevar un adecuado control sobre el inventario de los materiales a difundir y de donación, para lo cual se controlará la información que se manipule (ingresar, modificar, eliminar) a través de procesos y además de información almacenada en las respectivas tablas.

- **Bodega:** almacenará información básica sobre las encargadas del centro de difusión y documentación, como son; código que será la cedula de identidad de cada funcionaria, nombres completos, clave de acceso, cargo de la funcionaria.
- **Tipo de Transacción:** almacenará información detallada sobre para que se va a usar el material (biblioteca, charla, conferencia, consumo interno, difusión, estudios, eventos nacionales, eventos internacionales, foros, investigación, consultoría, etc.).
- **Tipos de Unidad:** los tipos de unidad son la presentación que tienen los materiales a difundir pueden ser docenas, medias docenas, centenas, unidades, etc.
- **Formato de tipo de documento:** se almacenará el formato del documento a usarse para las transacciones, pueden ser hojas de ruta, solicitud de entrega – recepción, email, verbal, documentos digitales, entre otros.
- **Productos:** Se almacenará los datos de los productos o materiales a

difundir, tendrá un código, una descripción, el costo, stock disponible y una alarma el cuál dirá cuando un material está pronto a agotarse.

- **Unidades Operativas (Ubicación):** Se pone el país, provincia o lugar a donde va el material.
- **Beneficiarios x Ubicación (Responsable):** Se almacenará la información necesaria sobre el responsable o quien recibe el material difundido o donado.
- **Usuario Final:** Se almacenará información necesaria sobre el usuario final, estos pueden ser: colegios, comités de usuarias, embajadas, empresas, escuelas, universidades, estados con sus diversas posibilidades, librerías, medios de comunicación, organizaciones de mujeres, personal de la institución, etc.

Además para llevar un adecuado inventario y control sobre los materiales de difusión y donación, se trabajará con las siguientes funciones o transacciones:

- **Ingreso:** Se realizará los procesos necesarios y controlados para el almacenamiento adecuado de la información sobre el material nuevo que se ingrese a stock o bodega.
- **Egresos:** Se realizarán los procesos necesarios y controlados para el almacenamiento adecuado de la información sobre el material que se envíe sean estos donados, difundidos o devueltos, según la opción que se tome.

- Realizar reportes estadísticos periódicos, estos reportes podrán ser creados a necesidad y propósito de cada usuario o circunstancia, los reportes a presentar constarán de datos estadísticos según el tipo de reporte que se escoja, también se podrá elegir el tipo de gráfico estadístico a presentar, además estos reportes ayudarán a desarrollar una mejor planificación y proyección sobre los materiales de difusión y donación, los reportes que serán mostrados son los siguientes:

- Reportes de Stock Disponible
- Reportes para resumen de Ingresos
- Ingresos Totales

- Objeto Receptor o usuario final
- Egresos Totales
- Egresos por Tipo de Transacción
- Acumulado por Producto
- Movimientos de Producto
- Movimientos por Destinatario

- Se deberá desarrollar y adecuar un módulo de Planificación sobre los materiales de difusión y donación, el cuál apoye a las funcionarias del área a realizar con anticipación pedidos también tener información necesaria sobre valores planificados y valores usados o ejecutados en el área de documentación y difusión, esta información junto con los reportes que se obtiene en el sistema serán usados por las encargadas del área para realizar una adecuada planificación de las actividades y los valores a usarse en el siguiente año. El módulo deberá llevar un seguimiento de las actividades, valores planificados y ejecutados en la planificación del área también se ingresaran los objetivos, metas y las actividades con sus respectivos valores, responsables y contrapartes.

Además el módulo permitirá manejar un listado con datos personales de las instituciones que aportan con material al CONAMU, para poder en un futuro saber el apoyo obtenido a lo largo de un año o un determinado tiempo.

- Se desarrollará un módulo para poder obtener reportes con datos, gráficos y análisis estadísticos obtenidos con la información almacenada mediante consultas personalizadas, los reportes detallarán la información mediante indicadores, valores, porcentajes, gráficos estadísticos a elegir por la funcionaria, el módulo aportará a las encargadas del área una mayor apreciación de la situación de los materiales donados y difundidos, destinatarios finales, materiales de mayor y menor demanda, sitios geográficos de mayor y menor envío, organizaciones de mujeres con mayor adquisición de materiales, temas de consulta de mayor demanda. Los reportes tendrán la facilidad de ser exportados a formatos; Excel, Word, pdf, jpg. El módulo contará con una encuesta virtual, el cuál será llenado por los visitantes al centro de documentación y difusión, la encuesta se la hará con la finalidad de obtener información de cómo se lleva el área, condición y

actualidad de los materiales, atención de las funcionarias, temas de mayor consulta, entidades u organizaciones de mayor visita al área.

### **3.2. REQUERIMIENTOS NO FUNCIONALES**

Los requerimientos no funcionales del sistema se desarrollarán en base a los aspectos del mismo que sean visibles para las funcionarias encargadas del Centro de Documentación y Difusión del Consejo Nacional de las Mujeres y que no están relacionados de forma directa con el comportamiento funcional del sistema a desarrollarse e implementarse. Para los requerimientos no funcionales a implementar en el sistema se ha tomado en cuenta los siguientes factores:

Interfaz de usuario y factores humanos:

- Tipo de interfaz:
  - La interfaz final para las funcionarias será desarrollada para que su manejo sea amigable e intuitivo, listas o combos para selección de datos, campos para ingresar datos, grillas para visualización de datos almacenados, menús para la elección de opciones de una manera rápida y ágil, además llevará los colores de la institución y su respectivo logo.

De tal motivo la interfaz a desarrollar es una Interfaz Gráfica de Usuario (GUI), por la necesidad de trabajar con varios dispositivos como: teclado, mouse, impresora, scanner,
- Experiencia del usuario en otros sistemas de iguales características.
  - Es importante saber que experiencias tienen las funcionarias en el manejo de sistemas de similares características gráficas para que les sea de mayor facilidad el adaptarse al nuevo sistema.

Documentación requerida:

El manual de usuario será entregado a las encargadas del área de difusión y documentación también se entregará una copia del manual al departamento informático, el manual de usuario permitirá usar de una manera fácil el sistema. También se entregará un manual técnico o de mantenimiento y adicionalmente el diccionario de datos al departamento informático que permitirá a las personas

encargadas de sistemas realizar mantenimientos, actualizaciones y adecuaciones necesarias acorde a las necesidades y requerimientos surgidos en el momento sobre el sistema y la base de datos.

Consideraciones de Hardware:

- Tener en cuenta el tipo de PC, procesador, memoria, red que manejen las PC clientes como el servidor, los siguientes datos serian los que se debería tener en cuenta para un eficiente funcionamiento del sistema:

**Para Servidor:**

- Sistema Operativo Linux (puede ser cualquier plataforma)
- Sistema Operativo Windows 2000 Server
- Servidor Web – Apache 4.5
- Servidor de Base de Datos – Postgre SQL 8.4.0

**Requisitos mínimos del sistema**

- Con procesador Intel Pentium IV a 1.8 GHz o superior
- 512 MB de memoria RAM
- 8 GB de espacio de disco disponible
- Pantalla con resolución 800 x 600

**Configuración recomendada**

- PC con procesador Intel Pentium IV 2.4 GHz
- 1 GB de memoria RAM
- 40 GB de espacio de disco disponible
- Pantalla con resolución 1024 x 768

**Para Terminal:**

- Sistema Operativo Windows 98 o superior
- Navegador de Internet: Internet Explorer, Netscape Navigator, Firefox Mozilla.

**Requisitos mínimos del sistema**

- PC con procesador Intel Pentium IV de 1.8 Ghz o superior
- 256 Mb de memoria RAM
- 4 GB de espacio de disco disponible
- Pantalla con resolución 800 x 600 o superior

## RED

- Red LAN. Por la facilidad que se puede realizar cambios especialmente en el hardware.
- Capacidad de Transmisión de 100Mbps.
- Medio de Transmisión: Cable UTP Categoría 5 o superior.
- Topología de Red: la que este en uso en la institución.

Interoperabilidad:

- Permitir exportar la información que se obtenga en los reportes a varios formatos usados por las funcionarias para su respectivo análisis, estos formatos pueden ser: Excel, Word y PDF.

Factores de Calidad: Fiabilidad, disponibilidad, capacidad de respuesta.

- Tiempo de respuesta en ingreso y modificación de datos menor o igual a 3 segundos dentro de la red local (intranet).
- Tiempo de respuesta para consultas y respuestas con la mayor rapidez dependiendo de la consulta y los datos arrojados por las mismas.
- La información obtenida de las consultas deben ser fiables y estén siempre disponibles dependiendo del filtro de búsqueda y de las restricciones que se de a cada usuaria

Ambiente Físico: Condiciones climáticas, exposición a golpes, accidentes, descargas eléctricas.

- Por seguridad el computador que tiene la función de Servidor debe estar ubicado en un lugar adecuado llamado “Cuarto Frío” con la temperatura adecuada y conectado a un solo UPS o regulador de energía, también en un módulo exclusivo solo para el servidor para evitar golpes o estática entre computadoras.
- Las PC que sirven de terminal deben tener su respectivo regulador de energía para evitar daños por posibles descargas eléctricas

Recursos del sistema:

- Consumo mínimo de recursos teniendo activado solo los servicios necesarios
- Disponibilidad de recursos para que el sistema funcione correctamente.



## CAPÍTULO 4

### 4. ANÁLISIS Y DISEÑO DEL SISTEMA

A partir de las especificaciones de requerimientos del sistema se muestra los diagramas de Casos de uso y el diseño de la base de datos.

#### 4.1. ESPECIFICACIONES DE CASO DE USO

A partir del análisis se describirá qué es lo que el sistema debe hacer. Los Casos de Uso son qué hace el sistema desde el punto de vista del usuario, se describen un uso del sistema y cómo este interactúa con el usuario.

La especificación de caso de uso es una breve descripción en líneas generales de la funcionalidad del caso de uso, de los actores que intervienen y del entorno de invocación<sup>31</sup>.

##### 4.1.1. CARACTERÍSTICAS DE LOS USUARIOS

Hay tres tipos de usuario en esta aplicación: funcionarias, beneficiario, administradora.

Los usuarios del sistema son:

- Administradora es la encargada de controlar los procedimientos que se pueden realizar en el sistema.
- Funcionaria es la encargada de realizar los ingresos y egresos del material en el sistema, también de emitir los reportes referentes a los materiales tales como :
  - Reportes de Stock Disponible
  - Reportes para resumen de Ingresos
  - Ingresos Totales
  - Objeto Receptor o usuario final
  - Egresos Totales
  - Egresos por Tipo de Transacción
  - Acumulado por Producto

---

<sup>31</sup> Especificación casos de uso. [En línea].

<[http://www.cyta.com.ar/cta/proyectos/pi\\_sitio\\_htm/material/3\\_plantilla\\_especificacion\\_caso\\_Uso.doc](http://www.cyta.com.ar/cta/proyectos/pi_sitio_htm/material/3_plantilla_especificacion_caso_Uso.doc)>

- Movimientos de Producto
- Movimientos por Destinatario
- Beneficiario o Usuario Final son todas las entidades (colegios, universidades, personas naturales, organizaciones, etc.) que van a recibir (por donación ó difusión) el material (libros, folletos, trípticos, etc.) y la información referente a los temas solicitados.

#### 4.1.2. ROLES Y TAREAS DE LOS USUARIOS

Se ha identificado 3 roles que a continuación se detalla.

##### **Administradora: Tareas**

- Administrar cuentas de usuarios.
- Administrar roles a las cuentas de usuarios.
- Controlar el Kárdex, el stock máximo y mínimo de los materiales
- Tener acceso a los procedimientos del sistema.
- Analizar los reportes entregados por las funcionarias.
- Analizar resultados de encuesta virtual atención al cliente.

Factores Críticos de Éxito para el perfil:

- Estar dentro del rango idóneo de responsabilidad de su departamento asignado.

##### **Funcionaria: Tareas**

- Ingresar el material que se va a recibir.
- Entregar el material ya sea por donación o difusión.
- Emitir el detalle del material recibido.
- Recibir las solicitudes de adquisición de material de las entidades.
- Emitir el detalle del material entregado a las entidades,
- Realizar cambio ó devolución del material.
- Emitir los respectivos reportes por periodos de tiempo establecidos por el centro de difusión y documentación en acuerdo con el Consejo Nacional de las Mujeres; y por tiempo requerido por dicha Institución.

Factores Críticos de Éxito:

- Tener conocimientos de todas las actividades del Centro de Difusión y Documentación del Consejo Nacional de las Mujeres así como de los movimientos esenciales del material que se realiza en la institución.
- Tener conocimiento adecuado del uso y manejo del sistema a partir de su respectiva capacitación.

**Beneficiario o Usuario Final: Tareas**

- Realizar la petición del material especificando la cantidad y el tema de interés.
- Recibir el material solicitado y el detalle de entrega.
- Llenar la encuesta virtual con los datos requeridos en dicha encuesta (Opcional).

**Factores Críticos de Éxito:**

- Tener conocimientos básicos de computación.
- Manejar herramientas de Ofimática<sup>32</sup>.

#### **4.1.3. ESPECIFICACIÓN DE ESCENARIOS**

**Administradora**

- Verificar la Información correspondiente a los materiales existentes o agotados del Centro de Difusión y Documentación del Consejo Nacional de las Mujeres.
- Verificar los materiales ingresados por las funcionarias.
- Revisar las peticiones de materiales realizadas por los usuarios.
- Crear categorías para los materiales.

**Funcionaria**

- Verificar datos de la hoja de detalle.
- Generar el detalle de los materiales a los usuarios.
- Ingresar la entidad y destino del material.
- Generar el resumen de los materiales entregados a los usuarios.

---

<sup>32</sup> Se llama ofimática al equipamiento hardware y software usado para crear, coleccionar, almacenar, manipular y transmitir digitalmente la información necesaria en una oficina para realizar tareas y lograr objetivos básicos.

- Control de entrega y recepción de documentos según los datos del beneficiario.

### Beneficiario

- Realizar la consulta del tema de interés, sea esta por visita a las instalaciones, vía mail o telefónica.
- Realizar la solicitud de petición de materiales al Centro de Difusión y Documentación del Consejo Nacional de las Mujeres,
- Revisar el material adquirido.
- Realizar reclamos en caso de ser necesario.
- Registrar sus respectivos datos en la encuesta virtual (opcional).

## CASOS DE USO

### Caso de Uso: Ingresar al sistema

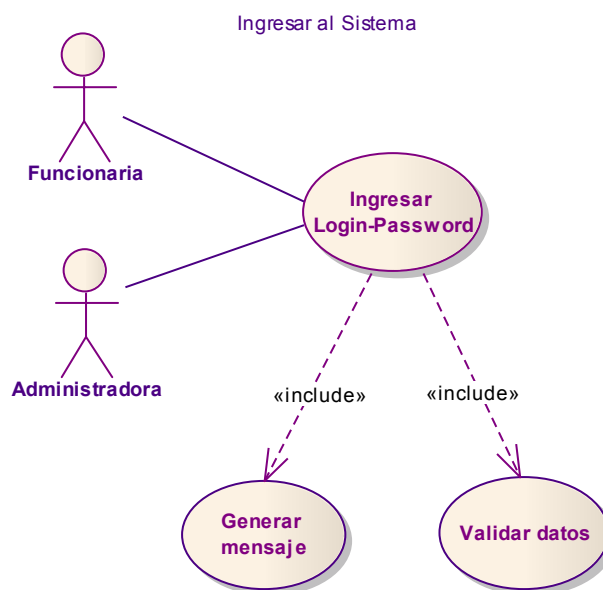


Figura 4.1. Caso de uso: Ingresar al sistema: Los Autores.

### Especificación de Caso de Uso: Ingresar funcionaria

<b>Caso de Uso</b>	Ingresar al Sistema
<b>Objetivo</b>	Ingresar al sistema por medio de la pantalla principal.
<b>Actores</b>	Usuaría.
<b>Precondiciones</b>	La usuaria debe disponer de su login y password para tener acceso al sistema.
<b>Acciones</b>	<ul style="list-style-type: none"> <li>• Ingresar al sistema por medio de su acceso directo o dirección URL.</li> </ul>

<b>Básicas</b>	<ul style="list-style-type: none"> <li>• El sistema despliega la pantalla de acceso donde solicita el login y password.</li> <li>• La usuaria debe registrar usuario y contraseña en los cuadros de texto provistos por la pantalla de acceso al sistema.</li> <li>• El sistema valida los datos que se han ingresado de dos formas posibles, presionando la enter luego de haber ingresado el password o través de un clic en el botón Aceptar.</li> <li>• El sistema valida los datos ingresados: <ul style="list-style-type: none"> <li>◦ Si estos son validos, el sistema muestra la pantalla del menú principal.</li> <li>◦ Si no lo son despliega mensaje de datos incorrectos.</li> </ul> </li> </ul>
<b>Pos condiciones</b>	<ul style="list-style-type: none"> <li>• El sistema muestra un mensaje en caso de que los datos ingresados al sistema estén erróneos en este caso sea usuario y password mal digitado o usuario no registrado.</li> <li>• Si los datos son correctos despliega la pantalla principal de sistema, habilitando las opciones del sistema según los roles asignados en la creación del usuario.</li> </ul>

Tabla 4.1. Especificación de casos de uso: Ingresar al sistema: Los Autores.

## Caso de Uso: Ingresar funcionaria

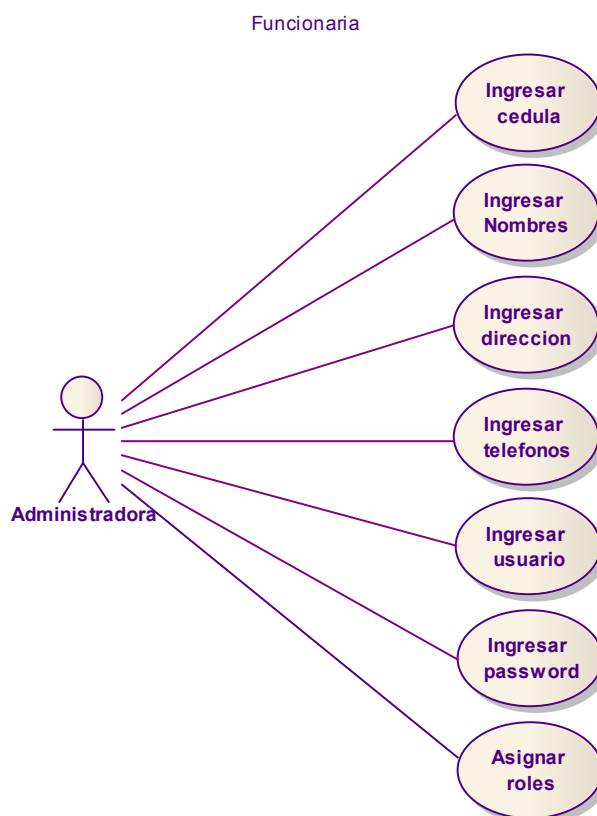


Figura 4.2. Caso de uso: Ingresar Funcionaria: Los Autores.

### Especificación de Caso de Uso: Ingresar funcionaria

<b>Caso de Uso</b>	Ingresar Funcionaria
<b>Objetivo</b>	Ingresar los datos que posee la funcionaria.
<b>Actores</b>	Administradora.
<b>Precondiciones</b>	La administradora posee el perfil con los permisos necesarios para ingresar y modificar a la información relacionada con funcionaria.
<b>Acciones Básicas</b>	<ul style="list-style-type: none"> <li>• Seleccionar la opción Catálogos de tablas básicas del menú principal, se despliega menú popup con las respectivas opciones.</li> <li>• Seleccionar opción Usuarios, el sistema desplegará la pantalla de ingreso de datos de Funcionaria.</li> <li>• Ingresar los datos de la funcionaria: cédula, nombres, teléfonos, dirección y correo electrónico.</li> <li>• Asignar el usuario con la respectiva contraseña.</li> <li>• Asignar el rol respectivo a la funcionaria.</li> <li>• Presionar en el botón guardar para que se almacene la información en la base de datos               <ul style="list-style-type: none"> <li>◦ Si los datos de cédula, nombre de usuario, usuario y password no han sido ingresados el sistema despliega un mensaje para que los ingrese</li> <li>◦ Si todos estos están ingresados, aun sin ingresar el resto de ítems de datos, el sistema almacenará la información de la funcionaria.</li> </ul> </li> </ul>
<b>Pos condiciones</b>	<ul style="list-style-type: none"> <li>• El sistema almacenará los datos de la funcionaria en la base de datos.</li> <li>• El sistema muestra la información de las funcionarias ingresadas al accionar en la opción Ver Funcionarias.</li> </ul>

Tabla 4.2. Especificación de casos de uso: Ingresar Funcionaria: Los Autores.

### Caso de Uso: Crear Tipo de transacción

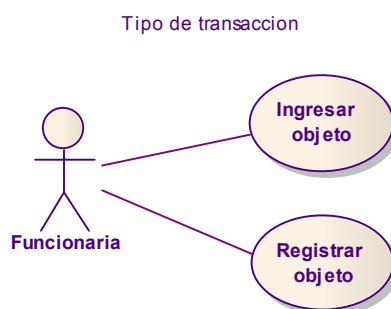


Figura 4.3. Caso de uso: Crear Tipo de transacción: Los Autores.

### Especificación de Caso de Uso: Crear Tipo de transacción

<b>Caso de Uso</b>	Crear Tipo de transacción.
<b>Objetivo</b>	Crear y añadir a la lista los tipos de transacción a usar.
<b>Actores</b>	Administradora - Funcionaria.
<b>Precondiciones</b>	La funcionaria posee el perfil con los permisos necesarios para crear Tipos de Transacciones.
<b>Acciones Básicas</b>	<ul style="list-style-type: none"> <li>• Desplegar la opción Catálogos de tablas básicas del menú principal.</li> <li>• Seleccionar opción Características, el sistema desplegará la pantalla para añadir un Tipo de Transacción.</li> <li>• Ingresar en el campo de texto la descripción del tipo de transacción.</li> <li>• Presionar en el botón guardar para que se almacene la información en la base de datos.</li> </ul>
<b>Pos condiciones</b>	<ul style="list-style-type: none"> <li>• El sistema almacenará los datos del tipo de transacción en la base de datos.</li> <li>• El sistema muestra la información de los tipos de transacción creados al accionar en la opción Ver Tipos de Transacción.</li> </ul>

Tabla 4.3. Especificación de casos de uso: Crear Tipos de Transacción: Los Autores.

### Caso de Uso: Crear Usuario final (Beneficiario)

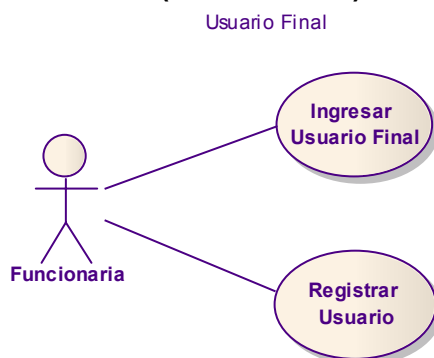


Figura 4.4. Caso de uso: Crear Usuario Final: Los Autores.

### Especificación de Caso de Uso: Crear Usuario final

<b>Caso de Uso</b>	Crear Usuario Final.
<b>Objetivo</b>	Crear y añadir a la lista los Usuarios Finales.
<b>Actores</b>	Administradora - Funcionaria.
<b>Precondiciones</b>	La funcionaria posee el perfil con los permisos necesarios para añadir Usuarios Finales.
<b>Acciones Básicas</b>	<ul style="list-style-type: none"> <li>• Desplegar la opción Catálogos de tablas básicas del menú principal.</li> <li>• Seleccionar opción Características, el sistema</li> </ul>

	<p>desplegará la pantalla para crear el Usuario Final.</p> <ul style="list-style-type: none"> <li>• Ingresar en el campo de texto la descripción del usuario final.</li> <li>• Presionar en el botón guardar para que se almacene la información en la base de datos.</li> </ul>
<b>Pos condiciones</b>	<ul style="list-style-type: none"> <li>• El sistema almacenará los datos del Usuario Final en la base de datos.</li> <li>• El sistema muestra la información de los usuarios finales añadidos al accionar en la opción Ver Usuario Final.</li> </ul>

Tabla 4.4. Especificación de casos de uso: Crear Usuario Final: Los Autores.

### Caso de Uso: Crear tipo de documento

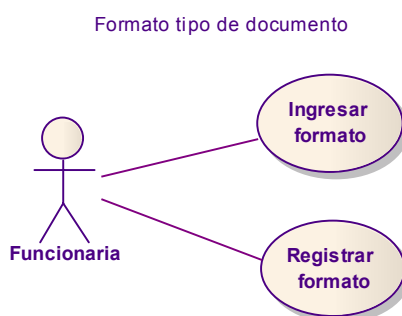


Figura 4.5. Caso de uso: Crear Tipo de Documento: Los Autores.

### Especificación de Caso de Uso: Crear tipo de documento

<b>Caso de Uso</b>	Crear Tipo de Documento.
<b>Objetivo</b>	Crear y añadir a la lista los Usuarios Finales.
<b>Actores</b>	Administradora - Funcionaria.
<b>Precondiciones</b>	La funcionaria posee el perfil con los permisos necesarios para crear formato de tipo de documento.
<b>Acciones Básicas</b>	<ul style="list-style-type: none"> <li>• Desplegar la opción Catálogos de tablas básicas del menú principal.</li> <li>• Seleccionar opción Productos, el sistema desplegará la pantalla Formato tipo de Documento.</li> <li>• Ingresar en el campo de texto la descripción del tipo de documento.</li> <li>• Presionar en el botón guardar para que se almacene la información en la base de datos.</li> </ul>
<b>Pos condiciones</b>	<ul style="list-style-type: none"> <li>• El sistema almacenará los datos del Tipo de documento en la base de datos.</li> <li>• El sistema muestra la información de los tipos de documentos añadidos al accionar en la opción Ver Tipo de Documentos.</li> </ul>

Tabla 4.5 Especificación de casos de uso: Crear Tipo de Documento: Los Autores.



## Caso de Uso: Ingresar Producto

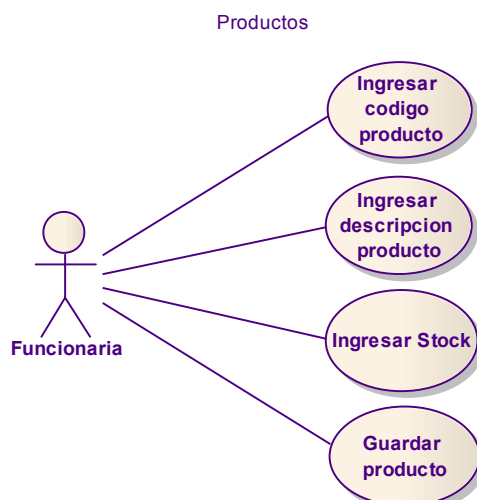


Figura 4.6. Caso de uso: Ingresar Producto: Los Autores.

## Especificación de Caso de Uso: Ingresar Producto

<b>Caso de Uso</b>	Ingresar Producto.
<b>Objetivo</b>	Realizar el ingreso de los productos.
<b>Actores</b>	Administradora - Funcionaria.
<b>Precondiciones</b>	La funcionaria posee el perfil con los permisos necesarios para realizar el ingreso de productos.
<b>Acciones Básicas</b>	<ul style="list-style-type: none"> <li>• Desplegar la opción Catálogos de tablas básicas del menú principal.</li> <li>• Seleccionar opción Productos, el sistema desplegará la pantalla de Producto.</li> <li>• Ingresar en los campos de texto los datos del producto: descripción, stock máximo, stock mínimo y existencias.</li> <li>• Elegir el tipo de unidad así como también el tipo de producto.</li> <li>• Presionar en el botón guardar para que se almacene la información en la base de datos.</li> </ul>
<b>Pos condiciones</b>	<ul style="list-style-type: none"> <li>• El sistema almacenará los datos del Producto en la base de datos.</li> <li>• El sistema muestra la información de los productos añadidos al accionar en la opción Ver Productos.</li> </ul>

Tabla 4.6. Especificación de casos de uso: Ingresar Producto: Los Autores.

### Caso de Uso: Crear tipo de unidad

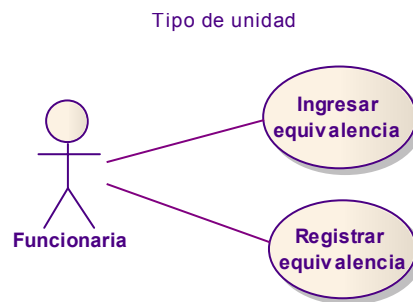


Figura 4.7. Caso de uso: Crear Tipo de Unidad: Los Autores.

### Especificación de Caso de Uso: Crear tipo de unidad

<b>Caso de Uso</b>	Crear Tipo de Unidad.
<b>Objetivo</b>	Crear el tipo de unidad para los productos.
<b>Actores</b>	Administradora - Funcionaria.
<b>Precondiciones</b>	La funcionaria posee el perfil con los permisos necesarios para crear Tipo de unidad.
<b>Acciones Básicas</b>	<ul style="list-style-type: none"> <li>• Desplegar la opción Catálogos de tablas básicas del menú principal.</li> <li>• Seleccionar opción Productos, el sistema desplegará la pantalla Tipo de Unidad.</li> <li>• Ingresar en los campos de texto los datos de Tipo de Unidad: código y la descripción.</li> <li>• Presionar en el botón guardar para que se almacene la información en la base de datos.</li> </ul>
<b>Pos condiciones</b>	<ul style="list-style-type: none"> <li>• El sistema almacenará los datos del Tipo de unidad en la base de datos.</li> <li>• El sistema muestra la información del tipo de unidad de los productos añadidos al accionar en la opción Ver Tipo de Unidad.</li> </ul>

Tabla 4.7. Especificación de casos de uso: Crear Tipo de Unidad: Los Autores.

### Caso de Uso: Crear tipo de producto

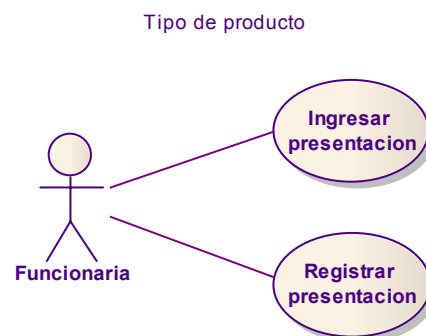


Figura 4.8. Caso de uso: Crear Tipo de Producto: Los Autores.

### Especificación de Caso de Uso: Crear tipo de producto

<b>Caso de Uso</b>	Crear Tipo de Producto.
<b>Objetivo</b>	Crear el tipo de producto para poder identificarlos ya sean estos revistas, trípticos o folletos, etc.
<b>Actores</b>	Administradora - Funcionaria.
<b>Precondiciones</b>	La funcionaria posee el perfil con los permisos necesarios para crear Tipo de producto.
<b>Acciones Básicas</b>	<ul style="list-style-type: none"> <li>• Desplegar la opción Catálogos de tablas básicas del menú principal.</li> <li>• Seleccionar opción Productos, el sistema desplegará la pantalla Tipo de Producto.</li> <li>• Ingresar en el campo de texto la descripción del tipo de producto.</li> <li>• Presionar en el botón guardar para que se almacene la información en la base de datos.</li> </ul>
<b>Pos condiciones</b>	<ul style="list-style-type: none"> <li>• El sistema almacenará los datos del Tipo de producto en la base de datos.</li> <li>• El sistema muestra la información del tipo de productos añadidos al accionar en la opción Ver Tipo de Producto.</li> </ul>

Tabla 4.8. Especificación de casos de uso: Crear Tipo de Producto: Los Autores.

### Caso de Uso: Ingresar proveedor

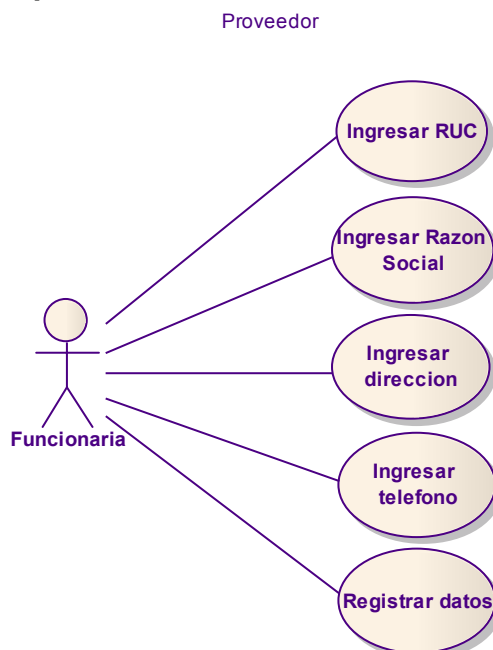


Figura 4.9. Caso de uso: Ingresar Proveedor: Los Autores.

### Especificación de Caso de Uso: Ingresar proveedor

<b>Caso de Uso</b>	Ingresar Proveedor.
<b>Objetivo</b>	Ingresar los datos de los proveedores de los productos didácticos.
<b>Actores</b>	Administradora - Funcionaria.
<b>Precondiciones</b>	La funcionaria posee el perfil con los permisos necesarios para ingresar proveedores.
<b>Acciones Básicas</b>	<ul style="list-style-type: none"> <li>• Desplegar la opción Catálogos de tablas básicas del menú principal.</li> <li>• Seleccionar opción Productos, el sistema desplegará la pantalla Proveedores.</li> <li>• Ingresar en los campos de texto los datos del Proveedor: RUC, razón social, dirección y teléfonos.</li> <li>• Presionar en el botón guardar para que se almacene la información en la base de datos.</li> </ul>
<b>Pos condiciones</b>	<ul style="list-style-type: none"> <li>• El sistema almacenará los datos del Proveedor en la base de datos.</li> <li>• El sistema muestra la información del tipo de Proveedor añadidos al accionar en la opción Ver Proveedores.</li> </ul>

Tabla 4.9. Especificación de casos de uso: Ingresar Proveedor: Los Autores.

### Caso de Uso: Ingresar destino geográfico

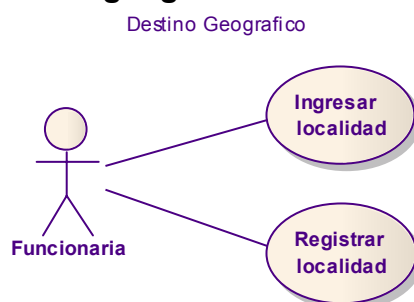


Figura 4.10. Caso de uso: Ingresar Destino Geográfico: Los Autores.

### Especificación de Caso de Uso: Ingresar destino geográfico

<b>Caso de Uso</b>	Ingresar Destino Geográfico.
<b>Objetivo</b>	Ingresar los nombres de los sitios o lugares de donde se recibirá o enviará los productos didácticos.
<b>Actores</b>	Administradora - Funcionaria.
<b>Precondiciones</b>	La funcionaria posee el perfil con los permisos necesarios para ingresar destino geográfico.
<b>Acciones Básicas</b>	<ul style="list-style-type: none"> <li>• Desplegar la opción Catálogos de tablas básicas del menú principal.</li> <li>• Seleccionar opción Transacción, el sistema</li> </ul>

	<p>desplegará la pantalla Destino Geográfico.</p> <ul style="list-style-type: none"> <li>• Ingresar en los campos de texto la descripción del Destino geográfico, sea esta ciudad o provincia.</li> <li>• Presionar en el botón guardar para que se almacene la información en la base de datos.</li> </ul>
<b>Pos condiciones</b>	<ul style="list-style-type: none"> <li>• El sistema almacenará los datos del Destino geográfico en la base de datos.</li> <li>• El sistema muestra la información del destino geográfico añadidos al accionar en la opción Ver Destino Geográfico.</li> </ul>

Tabla 4.10. Especificación de casos de uso: Ingresar Destino Geográfico: Los Autores.

### Caso de Uso: Ingresar responsable

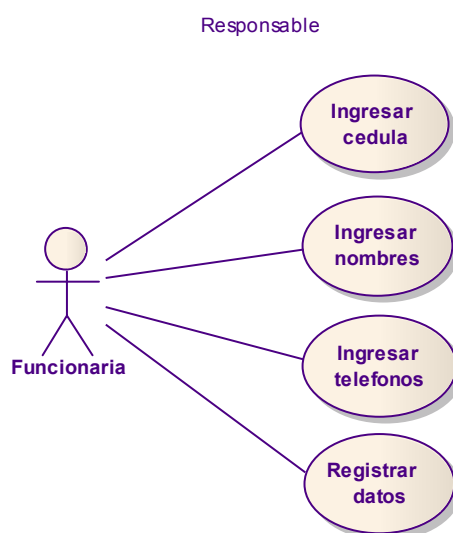


Figura 4.11. Caso de uso: Ingresar Responsable: Los Autores.

### Especificación de Caso de Uso: Ingresar responsable

<b>Caso de Uso</b>	Ingresar Responsable.
<b>Objetivo</b>	Ingresar los datos de la persona responsable para recibir o entregar los productos didácticos.
<b>Actores</b>	Administradora - Funcionaria.
<b>Precondiciones</b>	La funcionaria posee el perfil con los permisos necesarios para ingresar responsable.
<b>Acciones Básicas</b>	<ul style="list-style-type: none"> <li>• Desplegar la opción Catálogos de tablas básicas del menú principal.</li> <li>• Seleccionar opción Transacción, el sistema desplegará la pantalla Responsable.</li> <li>• Ingresar en los campos de texto los datos del Responsable: cedula, nombres, apellido y teléfono.</li> <li>• Presionar en el botón guardar para que se almacene la información en la base de datos.</li> </ul>

<b>Pos condiciones</b>	<ul style="list-style-type: none"> <li>El sistema almacenará los datos del responsable en la base de datos.</li> <li>El sistema muestra la información del responsable añadido al accionar en la opción Ver Responsables.</li> </ul>
------------------------	--

Tabla 4.11. Especificación de casos de uso: Ingresar Responsable: Los Autores.

### Caso de Uso: Crear tipo de usuario

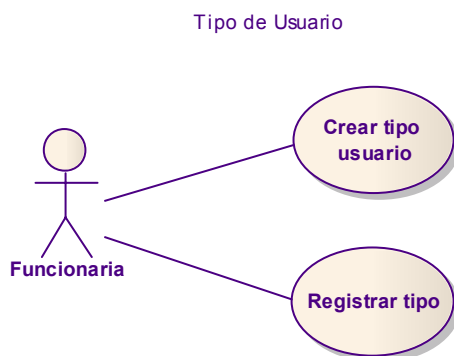


Figura 4.12. Caso de uso: Crear Tipo de Usuario: Los Autores.

### Especificación de Caso de Uso: Crear tipo de usuario

<b>Caso de Uso</b>	Crear Tipo de Usuario.
<b>Objetivo</b>	Crear tipos de usuarios que permitan diferenciar a los usuarios ya sean de primaria ,secundaria etc.
<b>Actores</b>	Administradora - Funcionaria.
<b>Precondiciones</b>	La funcionaria posee el perfil con los permisos necesarios para crear tipo de usuario.
<b>Acciones Básicas</b>	<ul style="list-style-type: none"> <li>Desplegar la opción Catálogos de tablas básicas, del menú principal.</li> <li>Seleccionar opción Evaluación Atención, el sistema desplegará la pantalla Tipo de Usuario.</li> <li>Ingresar en el campo de texto la descripción del tipo de usuario.</li> <li>Presionar en el botón guardar para que se almacene la información en la base de datos.</li> </ul>
<b>Pos condiciones</b>	<ul style="list-style-type: none"> <li>El sistema almacenará los datos del Tipo de usuario en la base de datos.</li> <li>El sistema muestra la información de los tipos de Usuario añadidos al accionar en la opción Ver Tipos de Usuario.</li> </ul>

Tabla 4.12. Especificación de casos de uso: Crear Tipo de Usuario: Los Autores.

### Caso de Uso: Crear tipo de servicio

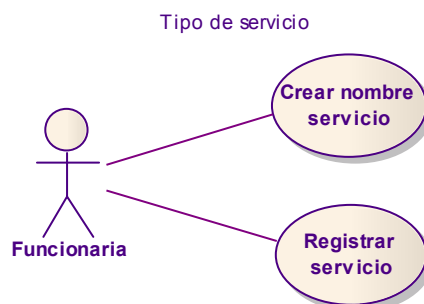


Figura 4.13. Caso de uso: Crear Tipo de Servicio: Los Autores.

### Especificación de Caso de Uso: Crear tipo de servicio

<b>Caso de Uso</b>	Crear Tipo de Servicio.
<b>Objetivo</b>	Crear tipos de servicio que permita indicar si la atención brindada fue mediante vía telefónica, visita a la biblioteca, etc.
<b>Actores</b>	Administradora - Funcionaria.
<b>Precondiciones</b>	La funcionaria posee el perfil con los permisos necesarios para crear tipo de servicio.
<b>Acciones Básicas</b>	<ul style="list-style-type: none"> <li>• Desplegar la opción Catálogos de tablas básicas del menú principal.</li> <li>• Seleccionar opción Evaluación Atención, el sistema desplegará la pantalla Tipo de Servicio.</li> <li>• Ingresar en el campo de texto la descripción del tipo de Servicio.</li> <li>• Presionar en el botón guardar para que se almacene la información en la base de datos.</li> </ul>
<b>Pos condiciones</b>	<ul style="list-style-type: none"> <li>• El sistema almacenará los datos del Tipo de servicio en la base de datos.</li> <li>• El sistema muestra la información de los tipos de Servicios añadidos al accionar en la opción Ver Tipos de Servicio.</li> </ul>

Tabla 4.13. Especificación de casos de uso: Crear Tipo de Servicio: Los Autores.

## Caso de Uso: Ingresar Organización



Figura 4.14. Caso de uso: Ingresar Organización: Los Autores.

## Especificación de Caso de Uso: Ingresar Organización

<b>Caso de Uso</b>	Ingresar Organización.
<b>Objetivo</b>	Ingresar los datos de las organizaciones ya sean estos Consejos, Ministerios, etc.
<b>Actores</b>	Administradora - Funcionaria.
<b>Precondiciones</b>	La funcionaria posee el perfil con los permisos necesarios para ingresar organizaciones.
<b>Acciones Básicas</b>	<ul style="list-style-type: none"> <li>• Desplegar la opción Catálogos de tablas básicas del menú principal.</li> <li>• Seleccionar opción Transacción, el sistema desplegará la pantalla Organización.</li> <li>• Ingresar en los campos de texto los datos de la Organización: nombre, dirección y teléfono.</li> <li>• Presionar en el botón guardar para que se almacene la información en la base de datos.</li> </ul>
<b>Pos condiciones</b>	<ul style="list-style-type: none"> <li>• El sistema almacenará los datos del Tipo de organización en la base de datos.</li> <li>• El sistema muestra la información de la organización añadida al accionar en la opción Ver Organizaciones.</li> </ul>

Tabla 4.14. Especificación de casos de uso: Ingresar Organización: Los Autores.



## Caso de Uso: Responder encuesta virtual

Encuesta virtual

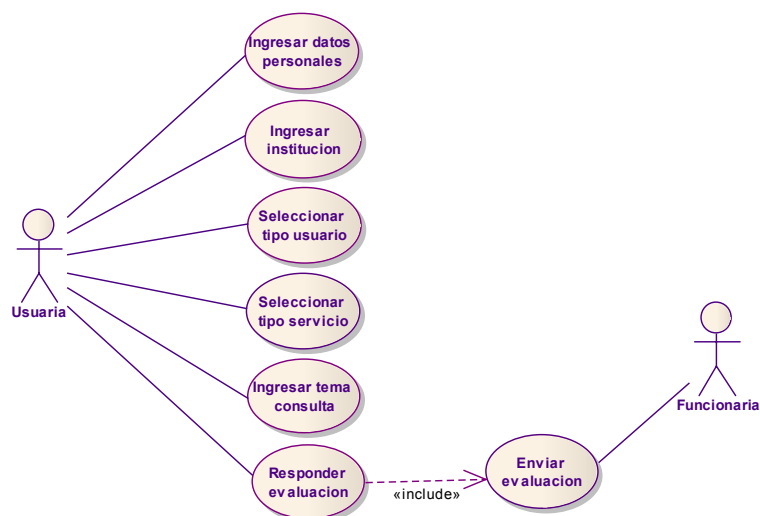


Figura 4.15. Caso de uso: Responder Encuesta Virtual: Los Autores.

## Especificación de Caso de Uso: Responder encuesta virtual

<b>Caso de Uso</b>	Responder Encuesta Virtual.
<b>Objetivo</b>	Obtener información de las usuarias para determinar los temas de interés más solicitados y tipo de usuarias mas frecuentes.
<b>Actores</b>	Funcionaria con perfil de usuario para activar la encuesta - Usuarías finales.
<b>Precondiciones</b>	Las usuarias poseen el acceso para acceder a la encuesta virtual y responder el formulario de la misma.
<b>Acciones Básicas</b>	<ul style="list-style-type: none"> <li>• Desplegar la opción Catálogos de tablas básicas del menú principal.</li> <li>• Seleccionar opción Ayuda, el sistema desplegará la pantalla Encuesta Virtual, la que permanecerá expuesta en el escritorio del computador para la facilidad de acceso de las usuarias.</li> <li>• Ingresar en los campos de texto los datos de la Encuesta Virtual: cedula, nombres, sexo, institución, mail, tema de consulta y teléfono.</li> <li>• Elegir la opción de tipo de usuario y tipo de servicio.</li> <li>• Presionar en el botón guardar para que se almacene la información en la base de datos.</li> </ul>
<b>Pos condiciones</b>	<ul style="list-style-type: none"> <li>• El sistema almacenará los datos de la Encuesta virtual en la base de datos.</li> <li>• El sistema muestra la información de las evaluaciones realizadas al accionar en la opción Ver Evaluaciones.</li> </ul>

Tabla 4.15. Especificación de casos de uso: Responder Encuesta Virtual: Los Autores.

## Caso de Uso: Realizar Ingresos

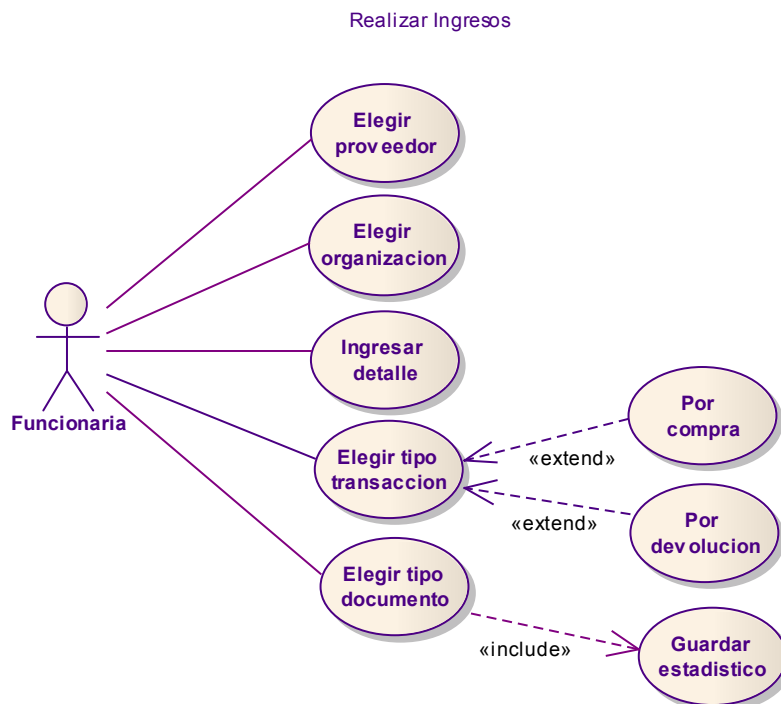


Figura 4.16. Caso de uso: Realizar Ingresos: Los Autores.

## Especificación de Caso de Uso: Realizar Ingresos

<b>Caso de Uso</b>	Realizar Ingresos.
<b>Objetivo</b>	Realizar Ingresos en la cabecera para almacenar un estadístico de productos.
<b>Actores</b>	Funcionaria.
<b>Precondiciones</b>	La Funcionaria posee el perfil con los permisos necesarios para realizar ingresos.
<b>Acciones Básicas</b>	<ul style="list-style-type: none"> <li>• Desplegar la opción Catálogos de tablas básicas del menú principal.</li> <li>• Seleccionar opción Transacciones, el sistema desplegará la pantalla Ingresos.</li> <li>• Elegir proveedor, organización, ingresar detalle, elegir tipo de documento, tipo de transacción.</li> <li>• Ingresar en el campo de texto el detalle de transacción.</li> <li>• Presionar en el botón guardar para que se almacene la información en la base de datos.</li> </ul>
<b>Pos condiciones</b>	<ul style="list-style-type: none"> <li>• El sistema almacenará los datos de los Ingresos en la base de datos.</li> <li>• El sistema muestra la información de los ingresos realizados al accionar en la opción Ver Ingresos.</li> </ul>

Tabla 4.16. Especificación de casos de uso: Realizar Ingresos: Los Autores.

## Caso de Uso: Realizar Egresos

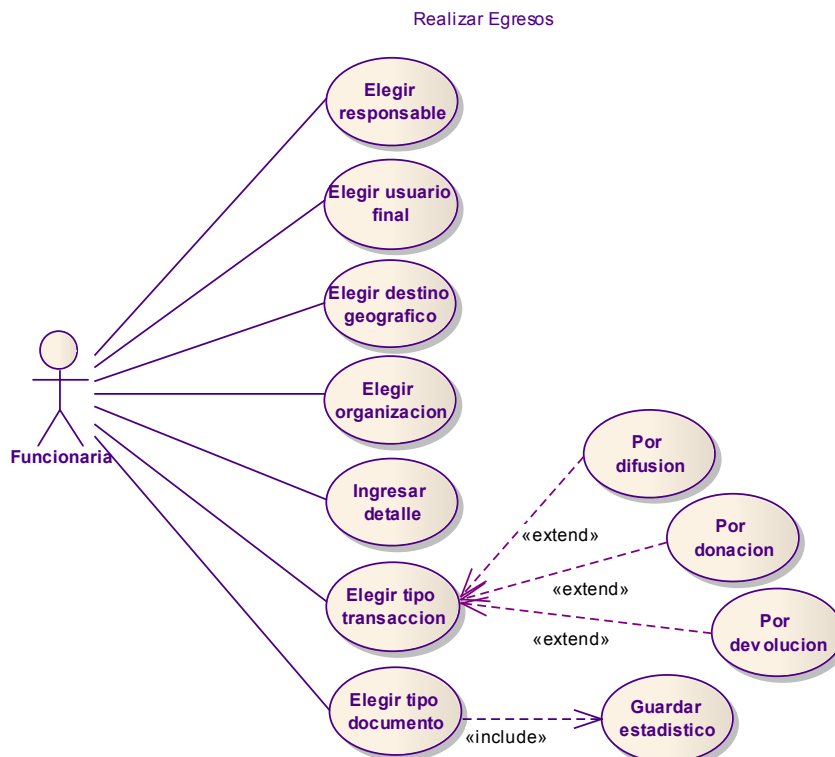


Figura 4.17. Caso de uso: Realizar Egresos: Los Autores.

### Especificación de Caso de Uso: Realizar Egresos

<b>Caso de Uso</b>	Realizar Egresos.
<b>Objetivo</b>	Realizar Egresos en la cabecera para almacenar un estadístico de productos.
<b>Actores</b>	Funcionaria.
<b>Precondiciones</b>	La Funcionaria posee el perfil con los permisos necesarios para realizar egresos.
<b>Acciones Básicas</b>	<ul style="list-style-type: none"> <li>• Desplegar la opción Catálogos de tablas básicas del menú principal.</li> <li>• Seleccionar opción Transacciones, el sistema desplegará la pantalla Egresos.</li> <li>• Elegir responsable, usuario final, ubicación, organización, tipo de documento, tipo de transacción.</li> <li>• Ingresar en el campo de texto el detalle de transacción.</li> <li>• Presionar en el botón guardar para que se almacene la información en la base de datos.</li> </ul>
<b>Pos condiciones</b>	<ul style="list-style-type: none"> <li>• El sistema almacenará los datos del Egreso en la base de datos.</li> <li>• El sistema muestra la información de los Egresos realizados al accionar en la opción Ver Egresos.</li> </ul>

Tabla 4.17. Especificación de casos de uso: Realizar Egresos: Los Autores.

## 4.2. DISEÑO DEL SISTEMA

### 4.2.1. ARCHIVOS O BASES DE DATOS

#### DISEÑO LÓGICO

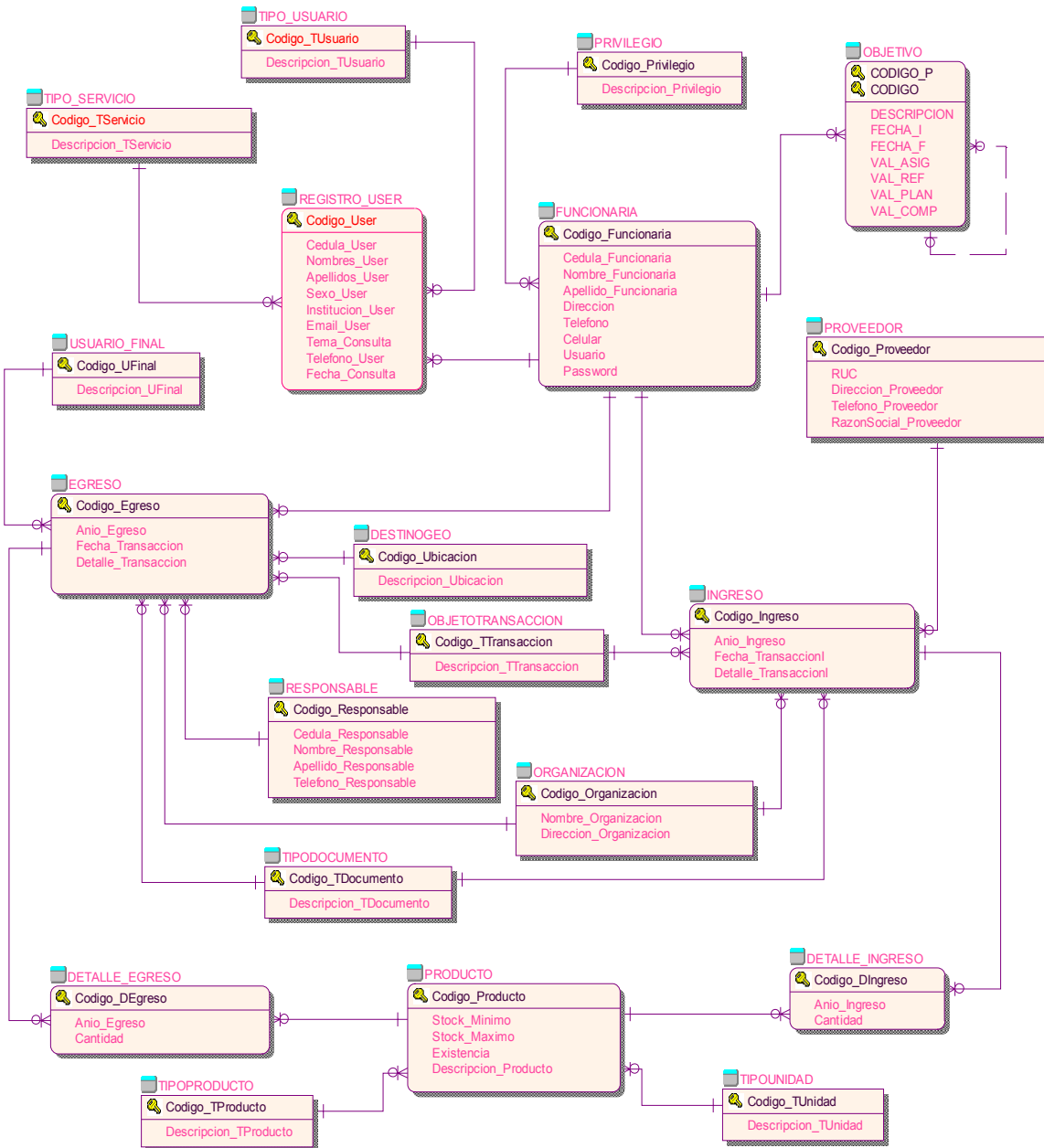


Figura4.18 . Base de Datos: Diseño Lógico: Los Autores.

## DISEÑO FÍSICO

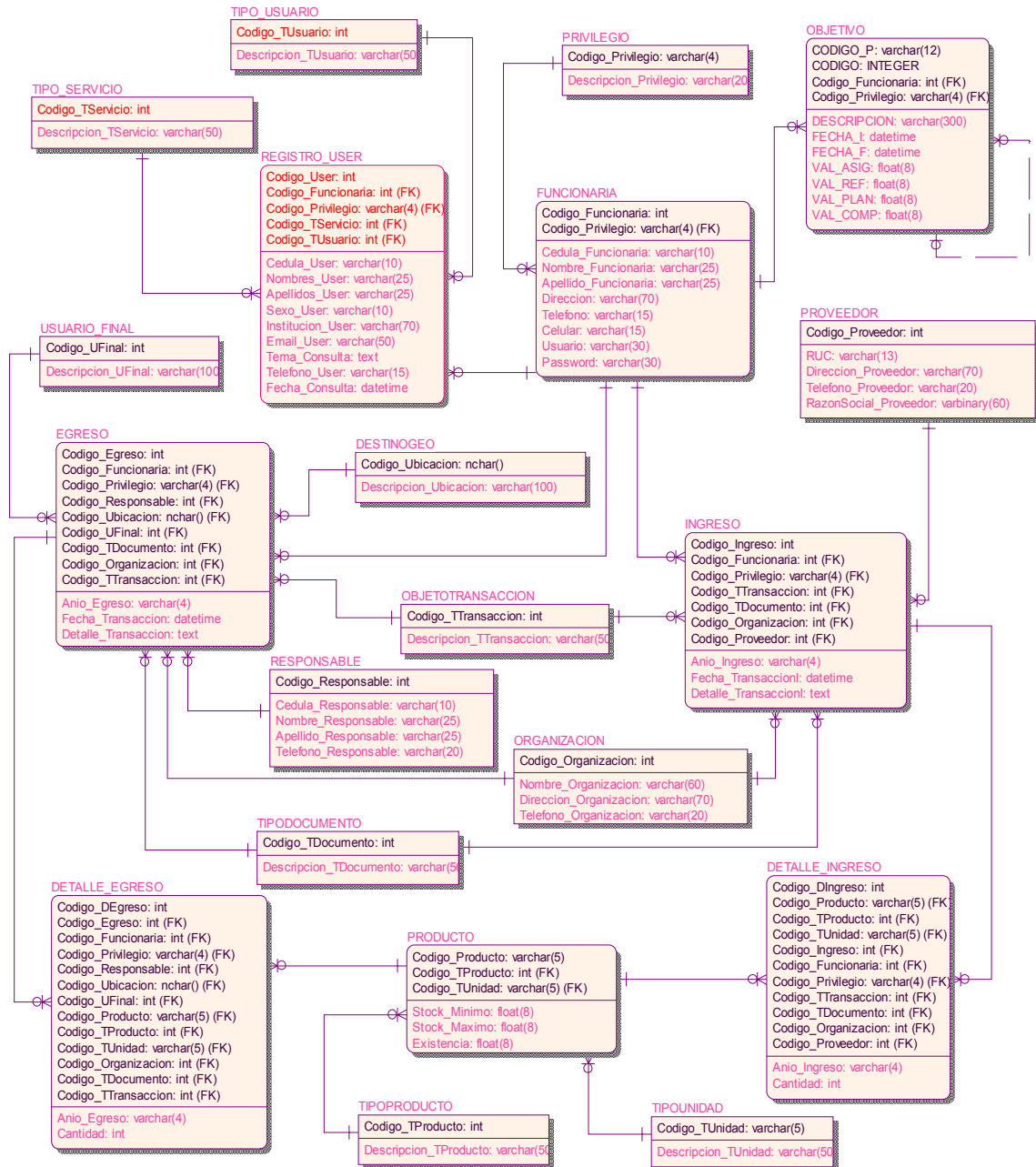


Figura4.19 . Base de Datos: Diseño Físico: Los Autores.

## **CAPÍTULO 5**

### **5. BASE DE DATOS POSTGRESQL**

#### **5.1. INTRODUCCIÓN AL POSTGRESQL**

Los sistemas de mantenimiento de Bases de Datos relacionales tradicionales (DBMS,s) soportan un modelo de datos que consisten en una colección de relaciones con nombre, que contienen atributos de un tipo específico. En los sistemas comerciales actuales, los tipos posibles incluyen numéricos de punto flotante, enteros, cadenas de caracteres, cantidades monetarias y fechas. Está generalmente reconocido que este modelo será inadecuado para las aplicaciones futuras de procesamiento de datos. El modelo relacional sustituyó modelos previos en parte por su "simplicidad espartana". Sin embargo, como se ha mencionado, esta simplicidad también hace muy difícil la implementación de ciertas aplicaciones. Postgres ofrece una potencia adicional sustancial al incorporar los siguientes cuatro conceptos adicionales básicos en una vía en la que los usuarios pueden extender fácilmente el sistema: clases, herencia, tipos, funciones.

Otras características aportan potencia y flexibilidad adicional: restricciones (Constraints), disparadores (triggers), reglas (rules), integridad transaccional.

Estas características colocan a Postgres en la categoría de las Bases de datos identificadas como objeto-relacionales. Éstas son diferentes de las referidas como orientadas a objetos, que en general no son bien aprovechables para soportar lenguajes de Bases de Datos relacionales tradicionales. Postgres tiene algunas características que son propias del mundo de las bases de datos orientadas a objetos.

##### **5.1.1. BREVE HISTORIA**

El Sistema Gestor de Bases de Datos Relacionales Orientadas a Objetos conocido como PostgreSQL (y brevemente llamado Postgres95) está derivado del paquete Postgres escrito en Berkeley. Con cerca de una década de desarrollo tras él, PostgreSQL es el gestor de bases de datos de código abierto más avanzado hoy en día, ofreciendo control de concurrencia multi-versión,

soportando casi toda la sintaxis SQL (incluyendo subconsultas, transacciones, y tipos y funciones definidas por el usuario), contando también con un amplio conjunto de enlaces con lenguajes de programación (incluyendo C, C++, Java, perl y python).<sup>33</sup>

En 1996, se hizo evidente que el nombre “Postgres95” no resistiría el paso del tiempo. Se elige un nuevo nombre, PostgreSQL, para reflejar la relación entre el Postgres original y las versiones más recientes con capacidades SQL. Al mismo tiempo, se hace que los números de versión partieran de la 6.0, volviendo a la secuencia seguida originalmente por el proyecto Postgres.

### 5.1.2. TERMINOLOGÍA

El sitio (o site) se puede interpretar como la máquina en la que está instalada Postgres. Dado que es posible instalar más de un conjunto de bases de datos Postgres en una misma máquina, este término denota cualquier conjunto concreto de programas binarios y bases de datos de Postgres instalados.

El superusuario de Postgres es el usuario llamado postgres que es dueño de los ficheros de la bases de datos y binarios de Postgres. Como superusuario de la base de datos, no le es aplicable ninguno de los mecanismos de protección y puede acceder a cualquiera de los datos de forma arbitraria. Además, al superusuario de Postgres se le permite ejecutar programas de soporte que generalmente no están disponibles para todos los usuarios. Tenga en cuenta que el superusuario de Postgres no es el mismo que el superusuario de Unix (que es conocido como *root*). El superusuario debería tener un identificador de usuario (*UID*) distinto de cero por razones de seguridad.

El administrador de la base de datos (*database administrator*) o DBA, es la persona responsable de instalar Postgres con mecanismos para hacer cumplir una política de seguridad para un site.

El DBA puede añadir nuevos usuarios y mantener un conjunto de bases de datos. El postmaster es el proceso que actúa como una puerta de control (clearing-house) para las peticiones al sistema Postgres. Las aplicaciones

---

<sup>33</sup> Manual de Usuario de PostgreSQL. [En línea].

<<http://postgresql.uci.cu/system/files/Manual%20del%20usuario%20de%20PostgreSQL.pdf>>. p. 2-3.

frontend<sup>34</sup> se conectan al postmaster, que mantiene registros de los errores del sistema y de la comunicación entre los procesos backend.

El postmaster puede aceptar varios argumentos desde la línea de órdenes para poner a punto su comportamiento. Sin embargo, el proporcionar argumentos es necesario sólo si se intenta trabajar con varios sitios o con uno que no se ejecuta a la manera por defecto.

El backend de Postgres (el programa ejecutable postgres real) lo puede ejecutar el superusuario directamente desde el intérprete de órdenes de usuario de Postgres (con el nombre de la base de datos como un argumento).

## 5.2. CONSTANTES

Hay tres tipos implícitos de constantes usadas Postgres: cadenas, enteros y números de coma flotante. Las Constantes también pueden ser especificadas con un tipo explícito, el cual puede tener una representación más adecuada y un manejo más eficiente.

### 5.2.1. CONSTANTES TIPO CADENA

Las cadenas son secuencias arbitrarias de caracteres ASCII limitadas por comillas simples (" ' ") SQL92 permite que las comillas simples puedan estar incluidos en una cadena tecleando dos comillas simples adyacentes. En Postgres las comillas simples deben estar precedidas por una contra barra ("\") para incluir una contra barra en una constante de tipo cadena, teclear dos contra barras. Los caracteres no imprimibles también deben incluir en la cadena precedidos de una contra barra ('\tab').

### 5.2.2. CONSTANTES TIPO ENTERO

Las constantes tipo entero son una colección de dígitos ASCII sin punto decimal. Los rangos de valores válidos van desde -2147483648 al +2147483647. Esto variará dependiendo del sistema operativo y la máquina host. El entero más largo puede ser especificado para int8 utilizando una notación de cadena SQL92 o una notación del tipo Postgres:

---

<sup>34</sup> Frontend y backend se relacionan con el principio y final de un proceso.



```
int8 '4000000000' - string style
'4000000000'::int8 - Postgres (historical) style
```

### 5.2.3. CONSTANTES TIPO PUNTO FLOTANTE

Las constantes de tipo Punto Flotante consta de una parte entera, punto decimal y una parte decimal o la notación científica con el siguiente formato:  $\{dig\}.\{dig\}$  [e  $[+-]\{dig\}$ ] donde *dig* es uno o más dígitos. Se puede incluir como mínimo un dig después del periodo y después de  $[+-]$  . Un exponente sin mantisa tiene una mantisa insertada a 1. No debe haber ningún carácter extra incluido en la cadena. Una constante de tipo punto flotante es del tipo float8. Para float4 se puede especificar explícitamente usando la notación de cadena de SQL92 o notación de tipo Postgres:

```
float4 '1.23' - string style
'1.23'::float4 - Postgres (historical) style
```

### 5.2.4. CONSTANTES DE TIPO DEFINIDO POR EL USUARIO

Una constante de un tipo arbitrario puede ser usado utilizando alguna de las siguientes notaciones:

```
type 'string'
'string'::type
CAST 'string' AS type
```

El valor de dentro de la cadena se pasa como entrada a rutina de conversión para el tipo llamado *type*. El resultado es una constante del tipo indicado. La tipología puede omitirse si no hay ambigüedad sobre el tipo de constate que debe ser, en este caso este está automáticamente forzado.

### 5.2.5. CONSTANTES DE TIPO ARRAY

El formato general de cualquier constante array es el siguiente:  $\{val1delimval2delim\}$  donde *delim* es el delimitador para el tipo almacenado en la clase pg\_type. Para los tipos preconstruidos, es el carácter coma. Un ejemplo de constante de tipo array es:  $\{\{1, 2,3\}, \{4, 5,6\}, \{7, 8,9\}\}$

Esta constante es de dos dimensiones, una array de 3 por 3 consiste en tres subarrays de enteros. Un elemento de un array individual puede y debe estar entre marcas delimitadoras siempre que sea posible para evitar problemas de ambigüedad con respecto a espacios en blanco iniciales.

### 5.3. SECUENCIAS

Una secuencia (sequence) se emplea para generar valores enteros secuenciales únicos y asignárselos a campos numéricos; se utilizan generalmente para las claves primarias de las tablas garantizando que sus valores no se repitan (normalmente utilizamos la definición de un campo serial, este tiene asociado una secuencia en forma automática).

Una secuencia es una tabla con un campo numérico en el cual se almacena un valor y cada vez que se consulta, se incrementa tal valor para la próxima consulta. Su sintaxis es la siguiente:

```
create sequence NOMBRESECUENCIA
start with VALORENTERO
increment by VALORENTERO
maxvalue VALORENTERO
minvalue VALORENTERO
cycle;
```

- La cláusula "start with" indica el valor desde el cual comenzará la generación de números secuenciales. Si no se especifica, se inicia con el valor que indique "minvalue".
- La cláusula "increment by" especifica el incremento, es decir, la diferencia entre los números de la secuencia; debe ser un valor numérico entero positivo o negativo diferente de 0. Si no se indica, por defecto es 1.
- "maxvalue" define el valor máximo para la secuencia. Si se omite, por defecto es 9223372036854775807.
- "minvalue" establece el valor mínimo de la secuencia. Si se omite será -9223372036854775808.
- La cláusula "cycle" indica que, cuando la secuencia llegue a máximo valor (valor de "maxvalue") se reinicie, comenzando con el mínimo valor ("minvalue") nuevamente, es decir, la secuencia vuelve a utilizar los

números. Si se omite, por defecto la secuencia se crea "nocycle", lo que produce un error si supera el máximo valor.

Si no se especifica ninguna cláusula, excepto el nombre de la secuencia, por defecto, comenzará en 1, se incrementará en 1, el mínimo valor será: -9223372036854775808, el máximo será: 9223372036854775807 y "nocycle".

## 5.4. OPERADORES

Cualquier operador predefinido o definido por el usuario puede usarse en SQL. Los paréntesis pueden usarse para agrupar arbitrariamente los operadores en expresiones.

Postgres proporciona un gran número de tipos de operadores. Estos operadores están declarados en el catálogo del sistema pg\_operator. Cada entrada en pg\_operator incluye el nombre del procedimiento que implementa el operador y las clases OIDs de los tipos de entrada y salida. Para ver todas las variantes del operador de concatenación de strings "||" se puede probar:

```
type 'string'
SELECT oprleft, oprright, oprresult, oprcode
FROM pg_operator WHERE oprname = '||';
oprleft|oprright|oprresult|oprcode
-----+-----+-----+-----
25| 25| 25|textcat
1042| 1042| 1042|textcat
1043| 1043| 1043|textcat
(3 rows)
```

Los usuarios pueden invocar a los operadores utilizando el nombre del operador de este modo:

```
select * from emp where salary < 40000;
```

De otra manera, los usuarios pueden llamar a las funciones que implementan los operadores directamente.

```
select * from emp where int4lt (salary, 40000);
```

Psql tiene un comando (\dd) para mostrar estos operadores.

## 5.5. PARÁMETROS

Un Parámetro se usa para indicar un parámetro en una función SQL. Típicamente este es el uso de la definición de la declaración de la función SQL. La forma con paréntesis es: *\$number*.

Ejemplo, consideramos la definición de la función, dept, como

```
CREATE FUNCTION dept (name)
RETURNS dept
AS 'select * from
dept where name=$1'
LANGUAGE 'sql';
```

### 5.5.1. EXPRESIONES FUNCIONALES

Una expresión funcional es el nombre de una función legal SQL, seguida por su lista de argumentos entre paréntesis:

```
function (a expr [, a expr ... ] )
```

Ejemplo, calcula la raíz cuadrada del salario de un empleado:

```
sqrt(emp.salary)
```

### 5.5.2. EXPRESIONES DE AGREGACIÓN

Una expresión de agregación representa la aplicación de una función de agregación a través de las filas seleccionadas por la consulta.

Una función de agregación reduce múltiples entradas a un solo valor de salida, como la suma o la media de la entrada. La sintaxis de la expresión de agregación es la siguiente:

```
aggregate_name (expression)
aggregate_name (ALL expression)
aggregate_name (DISTINCT expression)
aggregate_name ( * )
```

Donde *aggregate\_name* es la agregación previamente definida, y *expresión* es cualquier expresión que no contenga a su vez ninguna expresión de agregación. La primera forma de expresión de agregación llama a la agregación a través de todas las filas de entrada la expresión devuelve un valor no nulo.

La segunda forma es similar a la primera, pero ALL es por defecto.

La tercera forma llama a la agregación para todas las filas de entrada con valores distintos entre si y no nulo. La última forma llama a la agregación para cada una de las filas de entrada sean con valor nulo o no, generalmente sólo es útil para la agregación count(). Ejemplo, count(\*) devuelve el número total de filas de entrada; count(f1) devuelve el número de filas de entrada donde f1 no es nulo; count(distinct f1) devuelve el número de distintos valores no nulos de f1.

### 5.5.3. LISTA OBJETIVO

Una Lista Objetivo es una lista de uno o más elementos separados por comas y entre paréntesis, cada una debe ser de la forma:

*a\_expr* [ AS *result\_attname* ]

Donde *result\_attname* es el nombre del atributo que ha sido creado.

Si *result\_attname* no está presente, entonces *a\_expr* debe contener sólo un nombre de atributo el cual se asumirá como el nombre del campo resultado. Sólo se usa el nombrado por defecto en Postgres si *a\_expr* es un atributo.

### 5.5.4. CALIFICADORES

Un calificador consiste en cualquier número de cláusulas conectadas por operadores lógicos: NOT, AND y OR.

Una cláusula es un *a\_expr* que se evalúa como un boolean sobre el conjunto de instancias.

## 5.6. INSTALACIÓN DEL POSTGRESQL

El instalador de PostgreSQL es basado en el Instalador de Windows, para estar familiarizado a más usuarios de Windows.

Se descarga la versión libre '8.4.1-1' del programa para Windows desde: <http://www.enterprisedb.com/products/pgdownload.do><sup>35</sup>.

Una vez descargado se lo tiene que ejecutar como administrador en el sistema. '*usuario administrador*' en Windows.

---

<sup>35</sup> Instalación Grafica de PostgreSQL. [En línea]. <<http://www.postgresql-es.org/node/361>>

Basta con pulsar en el fichero exe descargado para empezar a ejecutar este programa de instalación. La pantalla que se muestra es la bienvenida al instalador de PostgreSQL. A partir de esta pantalla se tendrá que pulsar "Next" para avanzar al siguiente paso.



Figura 5.1. Instalación de PostgreSQL: Setup - PostgreSQL

En el siguiente paso se definirá el directorio donde se va a instalar todos los programas que vienen con esta versión de PostgreSQL. Se utilizará el valor por defecto que el programa nos sugiere. c:\\Program Files\\PostgreSQL\\8.4 en Windows XP.

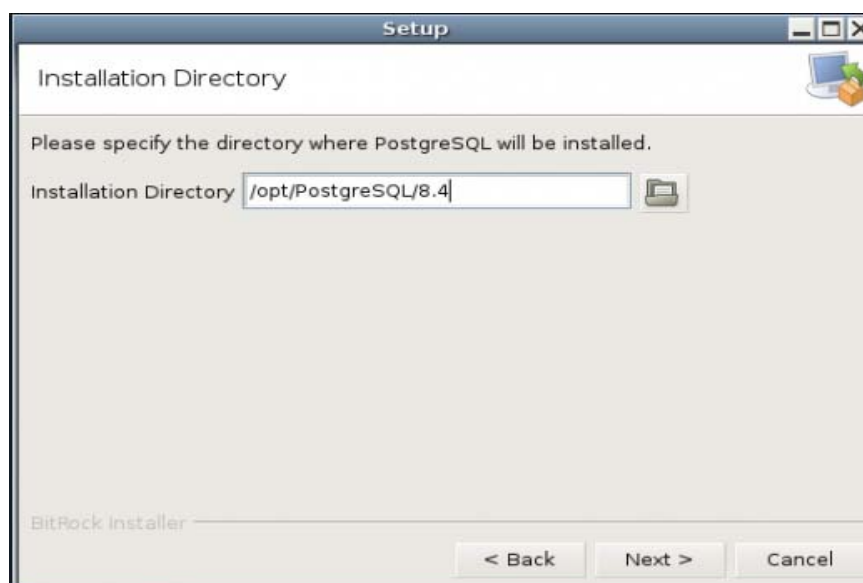


Figura 5.2. Instalación de PostgreSQL: Installation Directory

En este paso se tendrá que definir el directorio de datos en donde se crearán las bases de datos. `c:\\Program Files\\PostgreSQL\\8.4\\data` en Windows XP.



*Figura 5.3. Instalación de PostgreSQL: Data Directory*

En este paso se define una clave de acceso para el usuario administrador de la base de datos PostgreSQL. La clave definida en este paso será también la clave que se utilizará para la cuenta de servicio 'postgres' que se creará durante la instalación. Si esta cuenta ya existe, se tendrá que utilizar la clave que dicha cuenta tiene asignada. Esta clave es totalmente independiente de la clave de la cuenta de servicio 'postgres' en el sistema operativo.



*Figura 5.4. Instalación de PostgreSQL: Password*

En este paso se especificará el puerto que PostgreSQL utilizará para escuchar por conexiones. En este caso se dejará el valor por defecto, 5432.



Figura 5.5. Instalación de PostgreSQL: Port

En este paso se decide que 'locale' se quiere utilizar y si se desea instalar PL/pgSQL en la base de datos template1.

Se sugiere elegir "Locale: C" e instalar PL/pgSQL en template1. (template1 es una base de datos por defecto en PostgreSQL que se puede utilizar como plantilla cuando se crea bases de datos.

Todo lo que se instala y crea en template1, estará disponible en todas las bases de datos que usen template1 como plantilla).

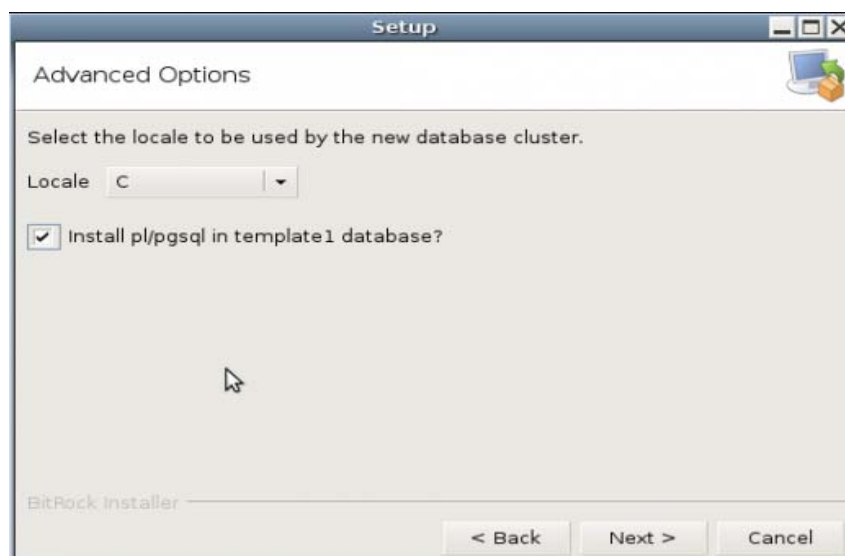
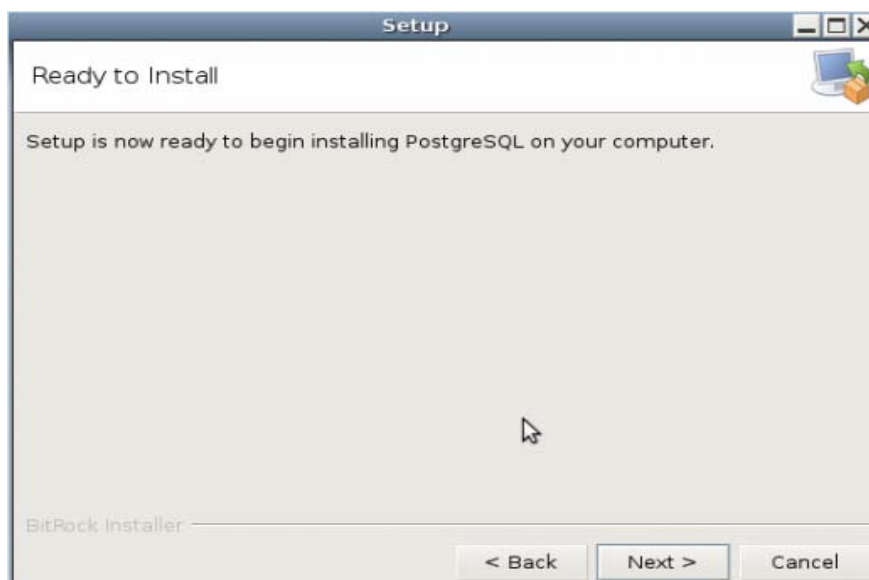


Figura 5.6. Instalación de PostgreSQL: Advanced Optiont

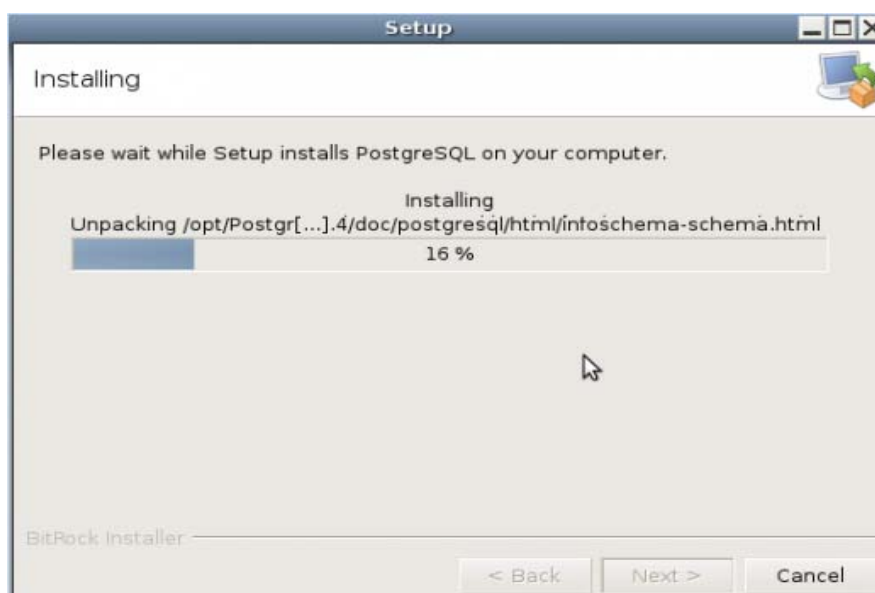


Una vez que se ha terminado con los pasos de configuración, el programa informa que está listo para empezar a instalar e inicializar la instalación.



*Figura 5.7. Instalación de PostgreSQL: Ready to install*

Se presiona por última vez "Next" y se espera a que el programa termine de instalar todo el contenido.



*Figura 5.8. Instalación de PostgreSQL: Installing*

La instalación tomara un intervalo de tiempo y continuará hasta completar su totalidad. Una vez terminada la instalación, se podra salir del instalador al presionar "Finish".



Figura 5.9. Instalación de PostgreSQL: Install

En este último paso el instalador da la opción de arrancar automáticamente un programa llamado "Stack Builder".

Este programa puede ser usado para instalar diversos programas adicionales, tanto programas libres como algunos distribuidos por EnterpriseDB.



Figura 5.10. Instalación de PostgreSQL: Completing

Stack Builder se puede correr al final de la instalación, o al inicio del Menú. Puede usarse para transmitir e instalar componentes adicionales y aplicaciones para

complementar cualquiera de sus instalaciones de PostgreSQL en la máquina local, o en un servidor remoto.

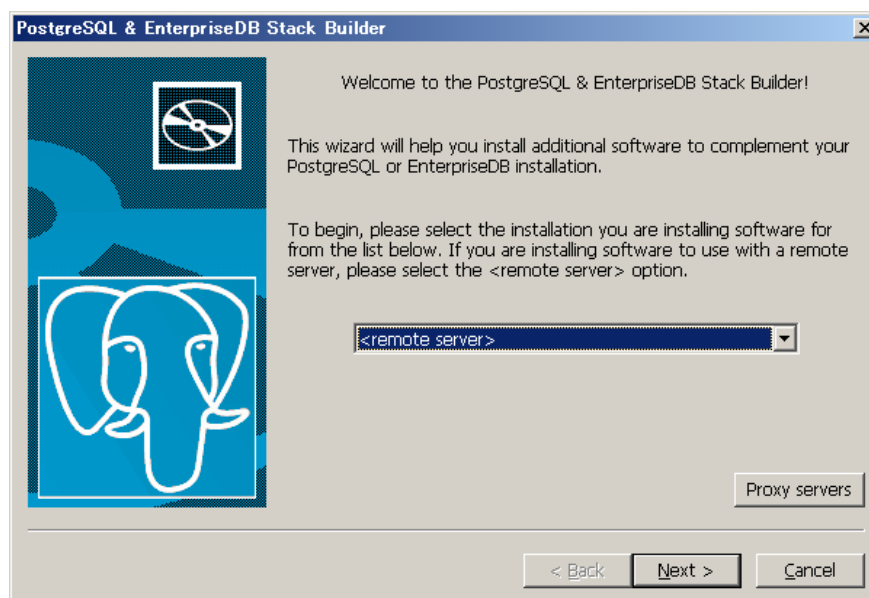


Figura 5.11. Instalación de PostgreSQL: Remote Server

Se selecciona los paquetes que se desean instalar. Si el paquete es dependiente de otro paquete este se seleccionará automáticamente.

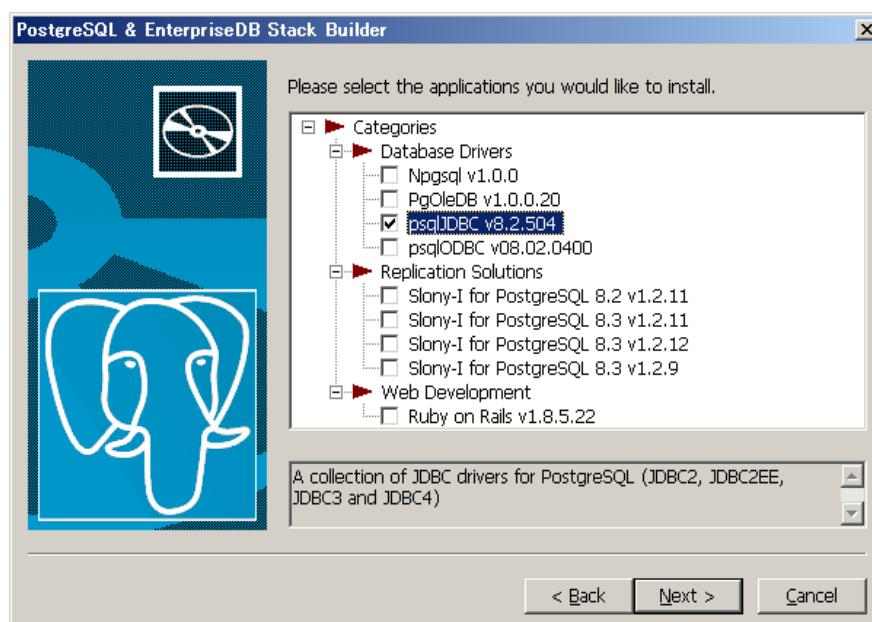


Figura 5.12. Instalación de PostgreSQL: Select the applications

Todos los paquetes y componentes seleccionados se proceden a instalar.

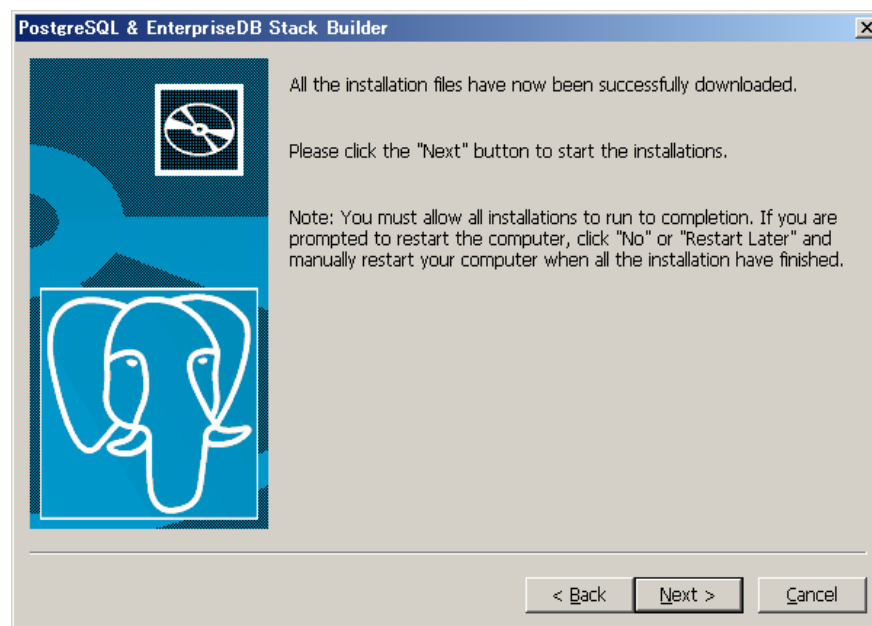


Figura 5.13. Instalación de PostgreSQL: Start the instalation

Iniciando la instalación.

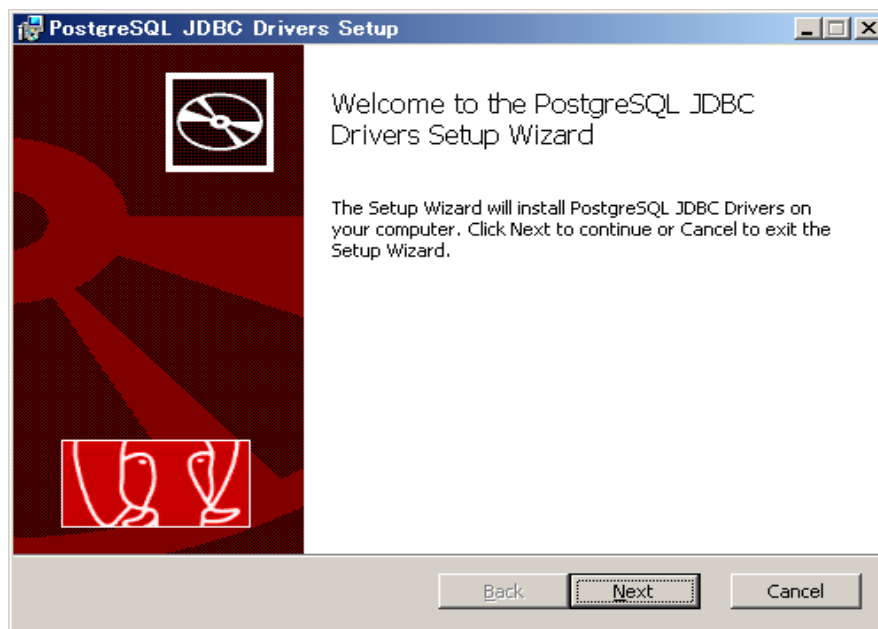


Figura 5.14. Instalación de PostgreSQL: JDBC Drivers

El proceso de la instalación ha sido completado, y el Stack Builder podría mostrar un resumen de todos los problemas encontrados en el momento de la instalación.

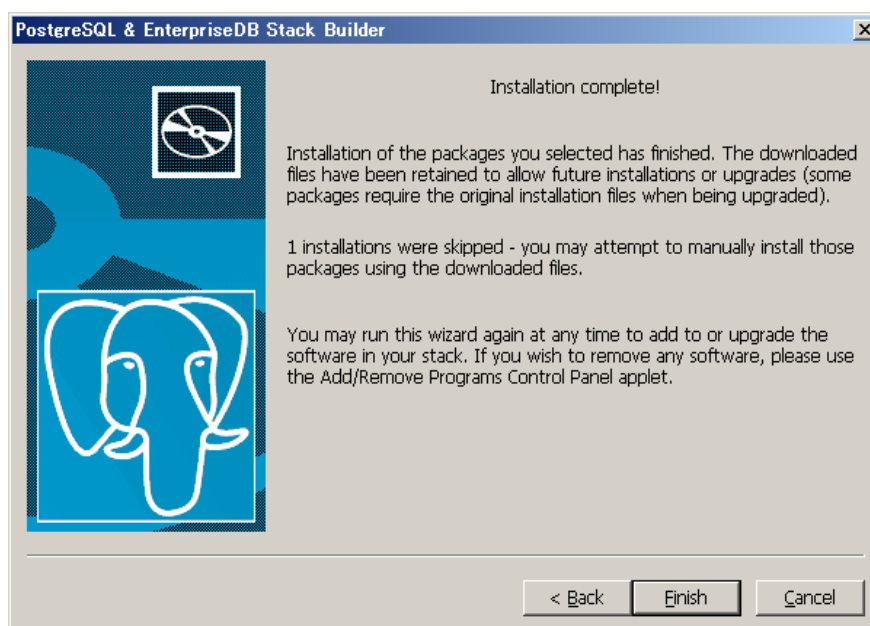


Figura 5.15. Instalación de PostgreSQL: Instalation complete

## 5.7. CONFIGURACIÓN DEL POSTGRESQL

A continuación se detallará sobre cómo configurar el propio entorno, de modo que pueda usar aplicaciones de interfaz de usuario.

El comportamiento de PostgreSQL en el sistema se puede controlar con tres ficheros de configuración que se encuentran en el directorio de datos donde se inicializará el cluster PostgreSQL (En este caso /var/pgsql/data).

Estos tres ficheros son:

***pg\_hba.conf***: Este fichero se utiliza para definir los diferentes tipos de accesos que un usuario tiene en el cluster.

***pg\_ident.conf***: Este fichero se utiliza para definir la información necesaria en el caso que se utilice un acceso del tipo ident en *pg\_hba.conf*.

***postgresql.conf***: En este fichero se podrá cambiar todos los parámetros de configuración que afectan al funcionamiento y al comportamiento de PostgreSQL en la máquina.

A continuación se explica los cambios más importantes que se puede hacer en algunos de estos ficheros.

***pg\_hba.conf***: Este fichero se utiliza para definir como, donde y desde que sitio un usuario puede utilizar el cluster PostgreSQL. Todas las líneas que empiecen

con el carácter # se interpretan como comentarios. El resto debe de tener el siguiente formato:

[Tipo de conexión][database][usuario][IP][Netmask][Tipo de autentificación] [opciones]  
 Dependiendo del tipo de conexión y del tipo de autentificación, [IP],[Netmask] y [opciones] pueden ser opcionales. El tipo de conexión puede tener los siguientes valores, local, host, hostssl y hostnossl. El tipo de método puede tener los siguientes valores, trust, reject, md5, crypt, password, krb5, ident, pam o ldap. Aquí unos ejemplos para comprender mejor como se puede configurar diferentes accesos al cluster PostgreSQL.

Para poner en producción los cambios en este fichero se tendrá que hacer que PostgreSQL vuelva a leerlo. Basta con un simple *'reload'* (/usr/local/bin/pg\_ctl -D /var/pgsql/data reload) desde la línea de comandos o con la función pg\_reload\_conf() como usuario postgres desde psql, el cliente PostgreSQL.

**postgresql.conf:** Los cambios que se realice en este fichero afectaran a todas las bases de datos que se tenga definidas en el cluster PostgreSQL. La mayoría de los cambios se pueden poner en producción con un simple *'reload'* (/usr/local/bin/pg\_ctl -D /var/pgsql/data reload), otros cambios necesitan que se arranque de nuevo el cluster (/usr/local/bin/pg\_ctl -D /var/pgsql/data restart).

A continuación se indica los parámetros más importantes que se debería cambiar si se empieza a usar PostgreSQL para un uso formal.

**max\_connections:** Número máximo de clientes conectados a la vez a la base de datos. Se debería de incrementar este valor en proporción al número de clientes concurrentes en el cluster PostgreSQL. Un buen valor para empezar es el 100:

max_connections = 100
-----------------------

**shared\_buffers:** Este parámetro es importantísimo y define el tamaño del buffer de memoria utilizado por PostgreSQL.

En un servidor dedicado se puede empezar con un 25% del total de la memoria. Nunca, más de 1/3 (33%) del total. Por ejemplo, en un servidor con 4Gbytes de memoria, podemos usar 1024MB como valor inicial.

```
shared_buffers = 1024MB
```

`work_mem`: Usada en operaciones que contengan ORDER BY, DISTINCT, joins, etc. En un servidor dedicado se puede usar un 2-4% del total de la memoria si se tiene solamente unas pocas sesiones (clientes) grandes. Como valor inicial se puede usar 8 Mbytes.

```
work_mem = 8MB
```

`maintenance_work_mem`: Usada en operaciones de tipo VACUUM, ANALYZE, CREATE INDEX, ALTER TABLE, ADD FOREIGN KEY. Su valor dependerá mucho del tamaño de la base de datos. Por ejemplo, en un servidor con 4Gbytes de memoria, se puede usar 256MB como valor inicial.

```
maintenance_work_mem = 256MB
```

`effective_cache_size`: Parámetro usado por el 'query planner' del motor de bases de datos para optimizar la lectura de datos. En un servidor dedicado se puede empezar con un 50% del total de la memoria. Como máximo unos 2/3 (66%) del total. Por ejemplo, en un servidor con 4Gbytes de memoria, se puede usar 2048MB como valor inicial.

```
effective_cache_size = 2048MB
```

`checkpoint_segments`: Este parámetro es muy importante en bases de datos con numerosas operaciones de escritura (insert, update, delete). Se puede empezar con un valor de 64.

```
Checkpoint_segments=64
```

En grandes databases con muchos Gbytes de datos escritos se puede aumentar este valor hasta 128-256.

Es muy importante tener en cuenta que al aumentar los valores por defecto de muchos parámetros, se tendrá que aumentar los valores por defecto de algunos parámetros del kernel de el sistema<sup>36</sup>.

---

<sup>36</sup> Introducción a PostgreSQL-Configuración. [En línea]. <<http://www.linux-es.org/node/660>>

## CAPÍTULO 6

### 6. DIAGRAMAS

A continuación se muestra los siguientes diagramas que pertenecen al sistema.

#### 6.1. DIAGRAMAS DE INTERACCIÓN

Los Diagramas de Interacción son un subtipo de diagramas de comportamiento, que enfatiza sobre el flujo de control y de datos entre los elementos del sistema modelado<sup>37</sup>:

##### 6.1.1. DIAGRAMAS DE SECUENCIA

Diagrama que muestra la interacción de los objetos ordenados en una secuencia de tiempo. Muestra los objetos que participan en la interacción y la secuencia de los mensajes que se intercambian.

#### Diagrama de secuencia: Ingresar al Sistema

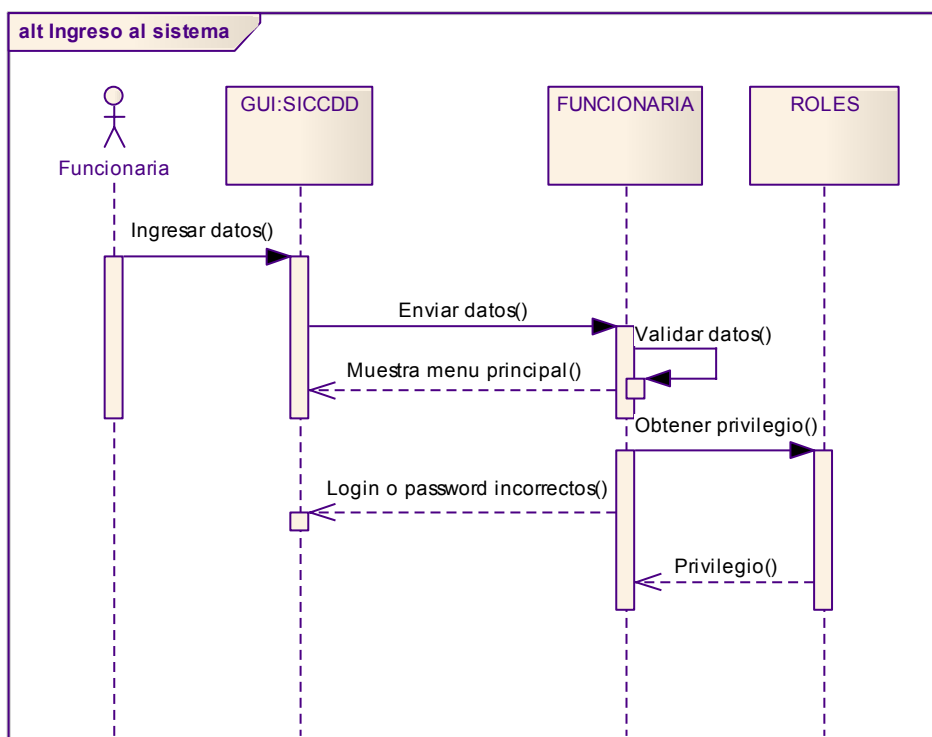


Figura 6.1. Diagrama de Secuencia: Ingresar al sistema: Los Autores.

<sup>37</sup> Lenguaje Unificado de Modelado. [En línea].  
<[http://es.wikipedia.org/wiki/Lenguaje\\_Unificado\\_de\\_Modelado](http://es.wikipedia.org/wiki/Lenguaje_Unificado_de_Modelado)>



### Diagrama de secuencia: Crear tipo de usuario

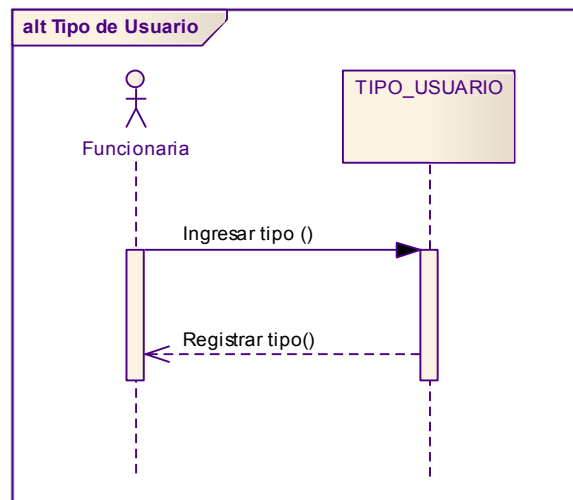


Figura 6.2. Diagrama de Secuencia: Crear Tipo de usuario: Los Autores.

### Diagrama de secuencia: Crear tipo de transacción

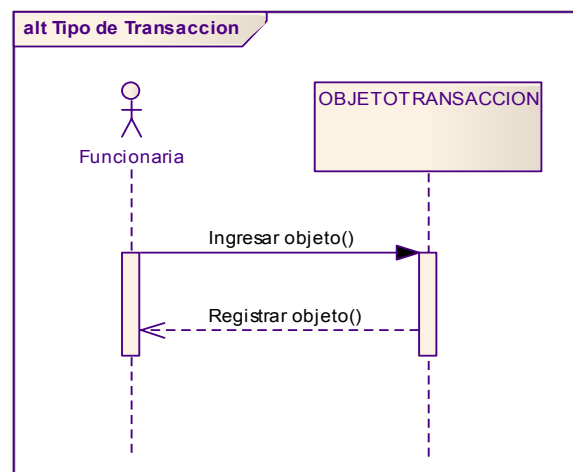


Figura 6.3. Diagrama de Secuencia: Crear Tipo de Transacción: Los Autores.

### Diagrama de secuencia: Ingresar usuario final

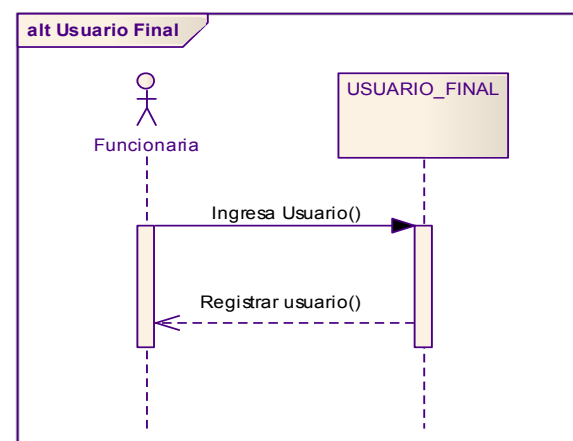


Figura 6.4. Diagrama de Secuencia: Ingresar Usuario Final: Los Autores.

### Diagrama de secuencia: Ingresar tipo de documento

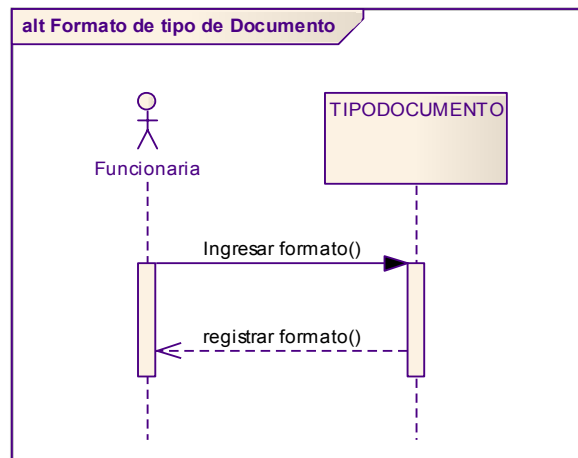


Figura 6.5. Diagrama de Secuencia: Ingresar Tipo de documento: Los Autores.

### Diagrama de secuencia: Ingresar productos

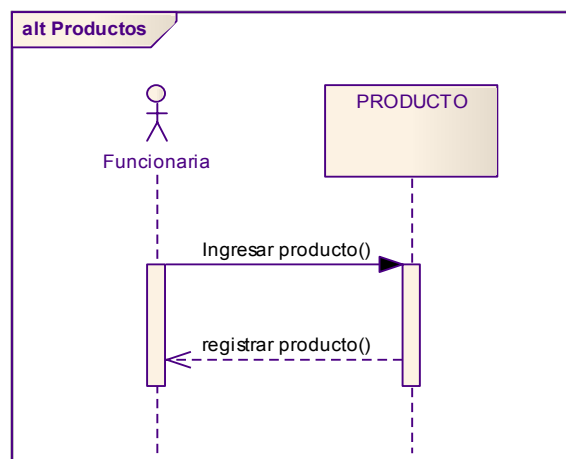


Figura 6.6. Diagrama de Secuencia: Ingresar Productos: Los Autores.

### Diagrama de secuencia: Ingresar tipo de unidad

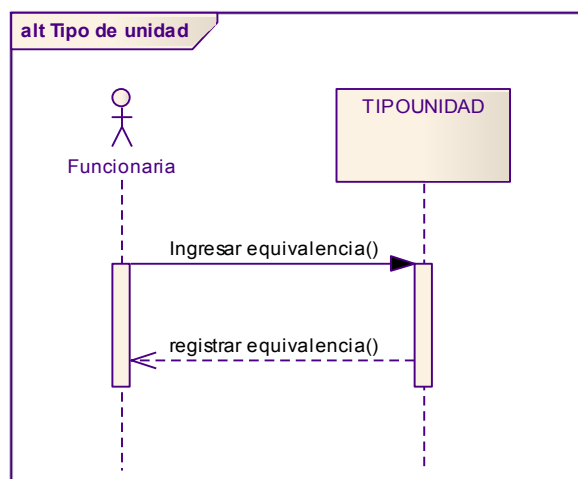


Figura 6.7. Diagrama de Secuencia: Ingresar Tipo de unidad: Los Autores.

### Diagrama de secuencia: Ingresar tipo de producto

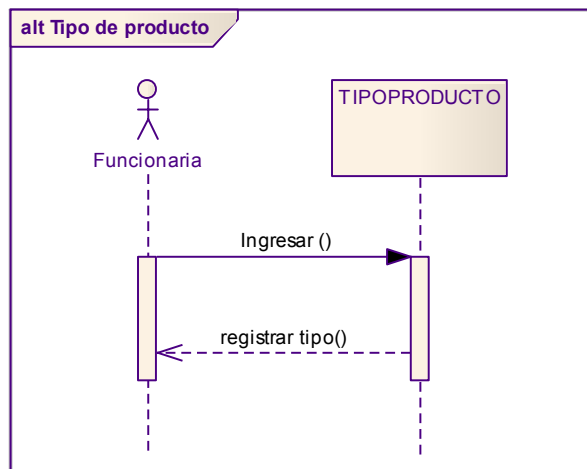


Figura 6.8. Diagrama de Secuencia: Ingresar Tipo de producto: Los Autores.

### Diagrama de secuencia: Ingresar proveedor

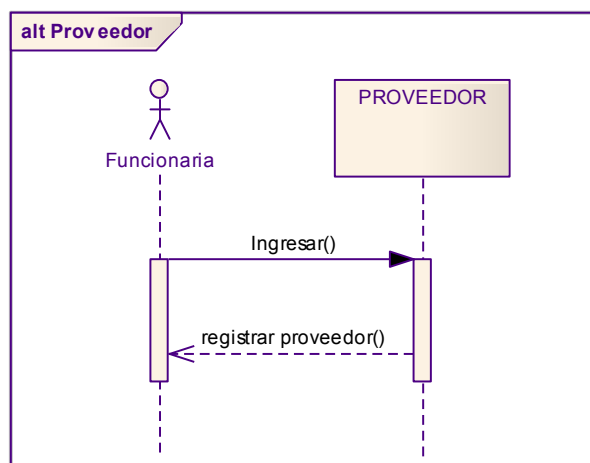


Figura 6.9. Diagrama de Secuencia: Ingresar Proveedor: Los Autores.

### Diagrama de secuencia: Ingresar destino geográfico

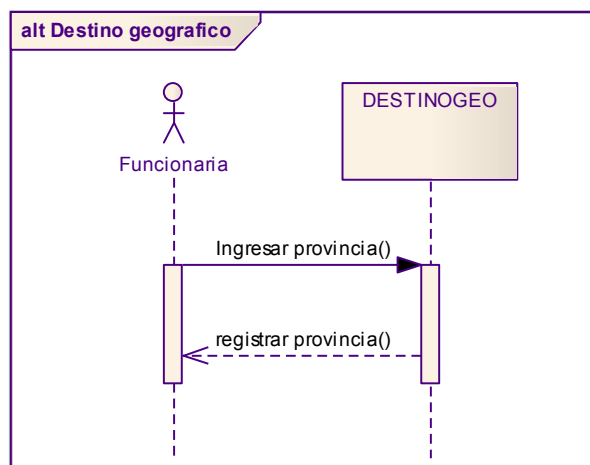


Figura 6.10. Diagrama de Secuencia: Ingresar Destino geográfico: Los Autores.

### Diagrama de secuencia: Ingresar responsable

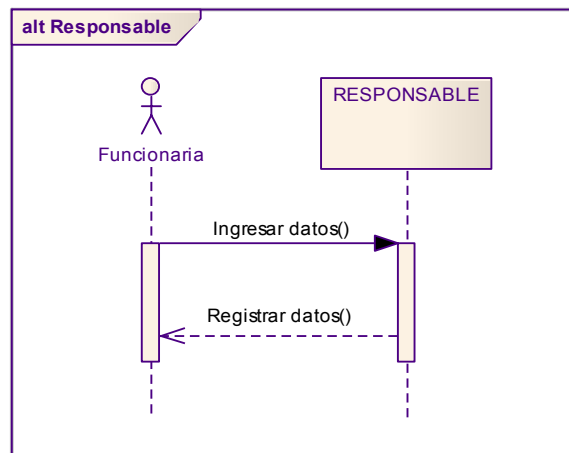


Figura 6.11. Diagrama de Secuencia: Ingresar Responsable: Los Autores.

### Diagrama de secuencia: Ingresar organización

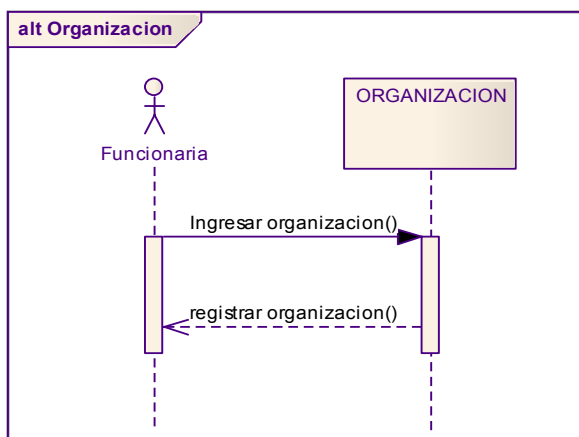


Figura 6.12. Diagrama de Secuencia: Ingresar Organización: Los Autores.

### Diagrama de secuencia: Crear tipo de servicio

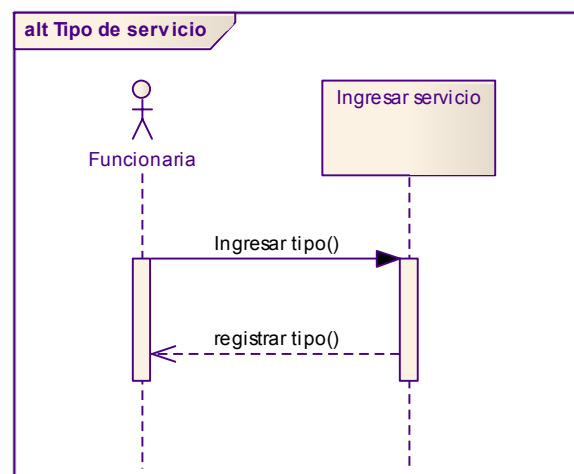


Figura 6.13. Diagrama de Secuencia: Crear Tipo de servicio: Los Autores.

### Diagrama de secuencia: Ingresar funcionaria

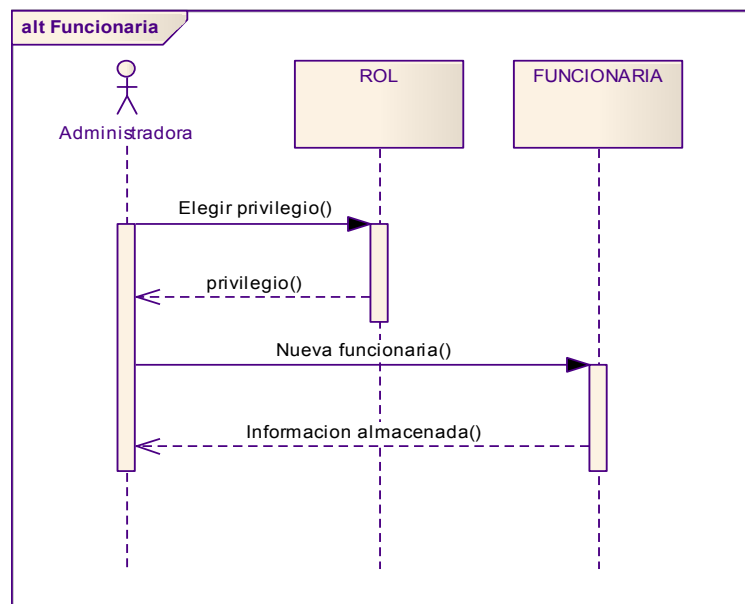


Figura 6.14. Diagrama de Secuencia: Ingresar Funcionaria: Los Autores.

### Diagrama de secuencia: Responder encuesta virtual

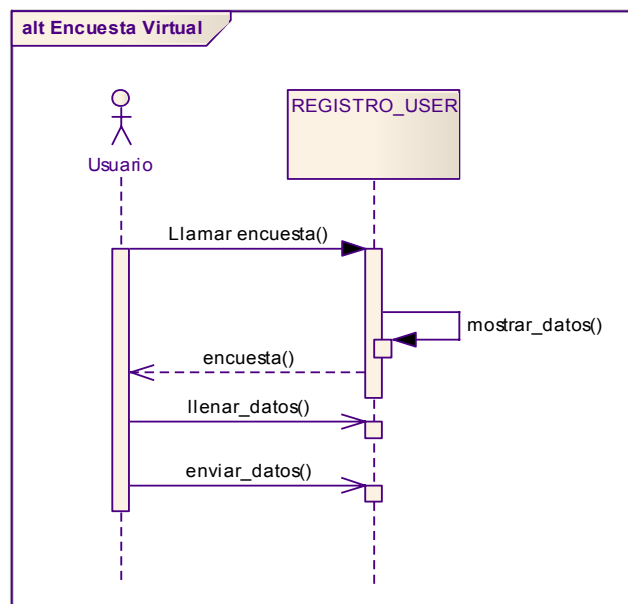


Figura 6.15. Diagrama de Secuencia: Responder Encuesta Virtual: Los Autores.

## 6.2. DIAGRAMA DE COMPORTAMIENTO

Los Diagramas de Comportamiento enfatizan en lo que debe suceder en el sistema<sup>38</sup>.

<sup>38</sup> Lenguaje Unificado de Modelado. [En línea].  
<[http://es.wikipedia.org/wiki/Lenguaje\\_Unificado\\_de\\_Modelado](http://es.wikipedia.org/wiki/Lenguaje_Unificado_de_Modelado)>

### 6.2.1. DIAGRAMA DE CASOS DE USO DEL NEGOCIO

Los diagramas de Casos de Uso del Negocio se usan para representar la funcionalidad proporcionada en conjunto por una organización.

Durante esta etapa, se contestan preguntas como: ¿Qué hace el sistema? o ¿Por qué se está construyendo el sistema? Estos diagramas no diferencian entre los procesos manuales y los automatizados, además muestran las interacciones entre los casos de uso del negocio y los actores del negocio<sup>39</sup>.

#### Verificar existencias en bodega

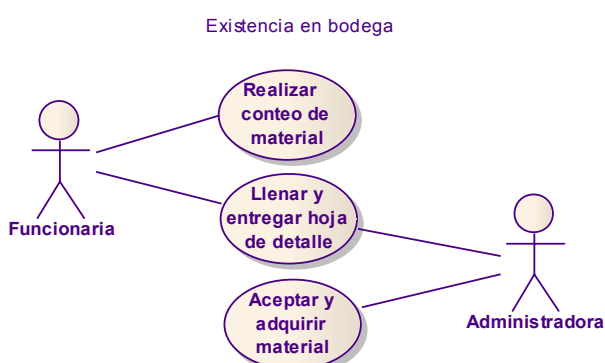


Figura 6.17.Caso de Uso del negocio: Verificar existencias en bodega: Los Autores.

En la Figura 6.17 la funcionaria realiza el conteo del material existente en bodega, hace la toma física del material procediendo a llenar las hojas de detalle. Las hojas de detalle son entregadas a la Administradora para que realice el análisis y haga la adquisición del material.

#### Recepción de mercadería del proveedor

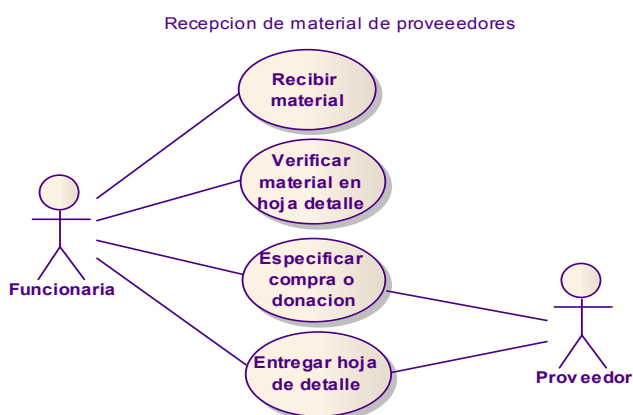


Figura 6.18.Caso de Uso del negocio: Recepción de mercadería del proveedor: Los Autores.

<sup>39</sup> Romero Moreno, Gesvin. Op. Cit. p. 18.

En la Figura 6.18 representa la recepción de la mercadería, donde el Proveedor entrega el material a la Funcionaria, la cual verifica los materiales con la hoja de detalle que el Proveedor emitió, en la que especifica si el material entregado es por compra o donación. Una vez que la Funcionaria verifica procede a recibir el material y a cancelar si este fuera por compra.

### Recepción de material de Organizaciones



Figura 6.19. Caso de Uso del negocio: Recepción de material de Organizaciones: Los Autores.

En la Figura 6.19 representa la recepción de la mercadería, donde el Proveedor entrega el material a la Funcionaria, la cual verifica los materiales con la hoja de detalle que el Proveedor emitió, en la que especifica si el material entregado es por donación o por devolución. Una vez que la Funcionaria verifica procede a recibir el material.

### Almacenar el material

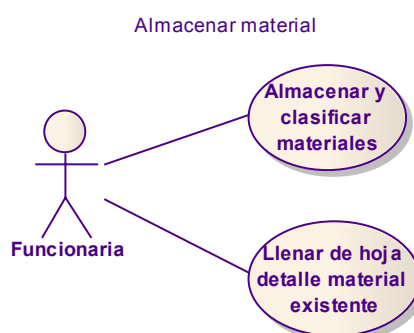


Figura 6.20. Caso de Uso del negocio: Almacenar el material: Los Autores.

En la Figura 6.20 representa almacenar el material, donde la Organización o Proveedor entrega el material a la Funcionaria y esta procede a clasificar y almacenar los materiales, después procede a llenar la hoja de detalle con el material existente.

### Entregar los materiales



Figura 6.21. Caso de Uso del negocio: Entregar material: Los Autores.

En la Figura 6.20 representa entregar el material al usuario final, donde la Funcionaria recibe y selecciona los materiales de la solicitud procediendo a entregar y enviar el material especificando en la hoja de detalle.

### 6.2.2. DIAGRAMA DE ACTIVIDADES

Los diagramas de actividades ilustran el flujo de funcionalidad en un sistema. Estos diagramas definen donde empieza y donde acaba el flujo del negocio y así conocer que actividades ocurren durante el flujo y en que orden ocurren. Una actividad es una tarea que se realiza durante el flujo de trabajo<sup>40</sup>.

<sup>40</sup> Romero Moreno, Gesvin. Op. Cit. p. 20.



### Verificar existencias en bodega

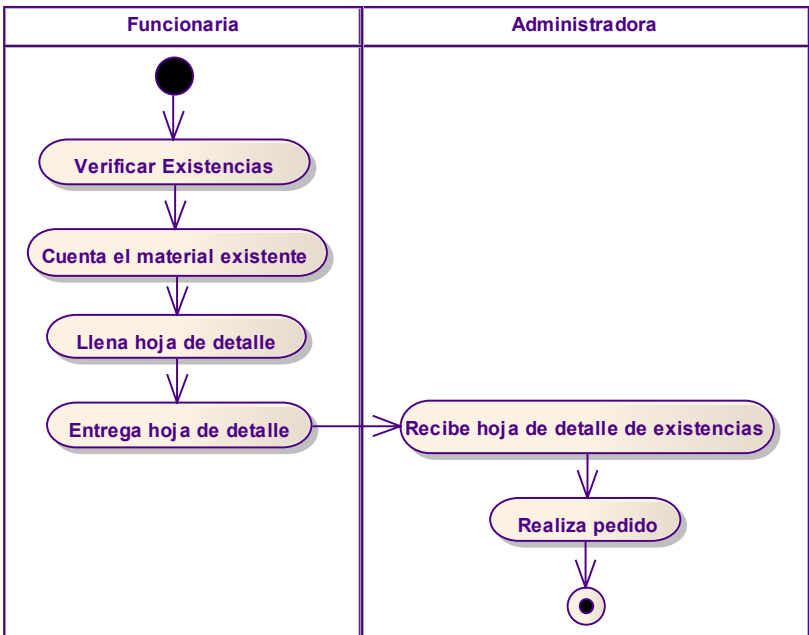


Figura 6.22. Diagrama de Actividades: Verificar existencias en bodega: Los Autores.

En el diagrama de actividades de la Figura 6.22 la Funcionaria para verificar las existencias, comprueba la existencia del material y procede a contar los materiales existentes, con los resultados obtenidos procede a llenar la hoja de detalle la cual será entregado a la Administradora quien esta facultada para hacer el análisis y realizar el pedido.

### Recepción de mercadería del proveedor

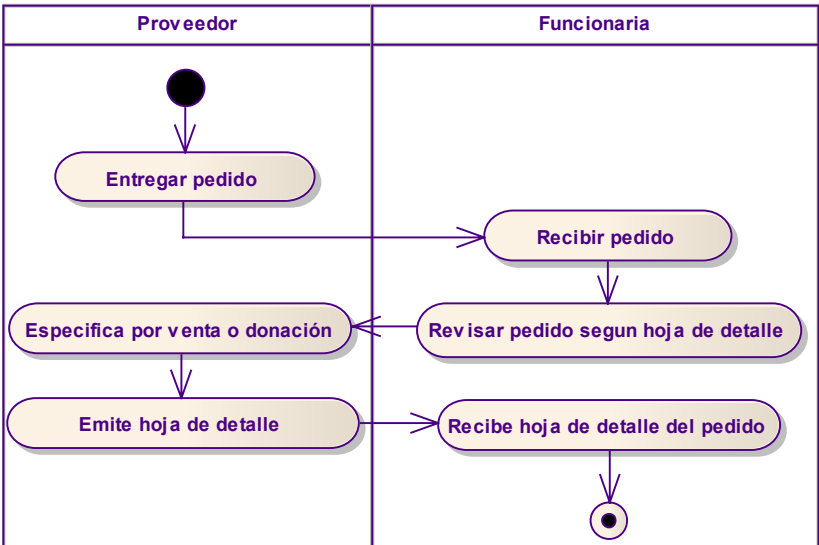


Figura 6.23. Diagrama de Actividades: Recepción de mercadería de proveedor: Los Autores.

En el diagrama de actividad de la Figura 6.23 al recibir el pedido del Proveedor la Funcionaria realiza las revisiones del pedido respaldándose en la hoja de detalle, el Proveedor especifica si el pedido a entregar se realiza por venta o donación, datos que se especifican en la hoja de detalle la misma que emite y entrega a la Funcionaria.

### Recepción de material de Organizaciones

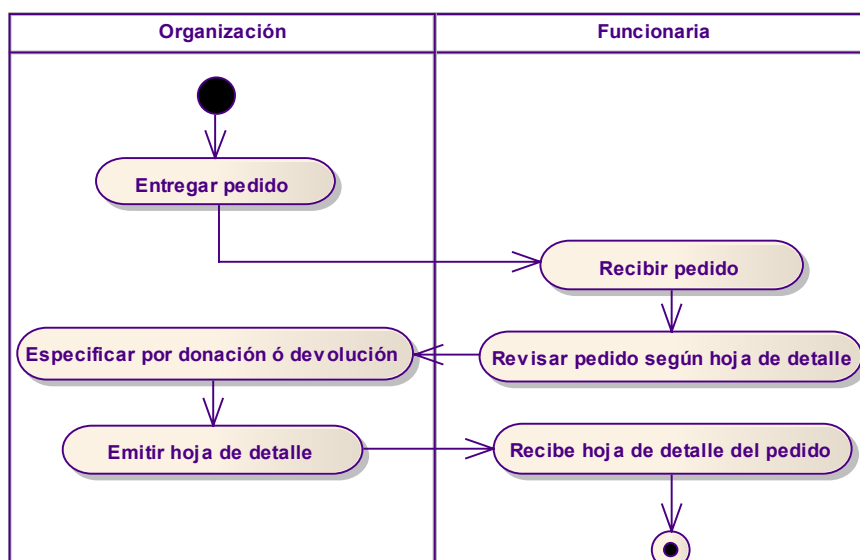


Figura 6.24. Diagrama de Actividades: Recepción de material de Organizaciones: Los Autores.

En el diagrama de actividad de la Figura 6.24 la Funcionaria al recibir el pedido realiza las revisiones respaldándose en la hoja de detalle, el Proveedor especifica si el pedido a entregar se realiza por donación o devolución, datos que se especifican en la hoja de detalle la misma que emite y entrega a la Funcionaria.

### Almacenar el material

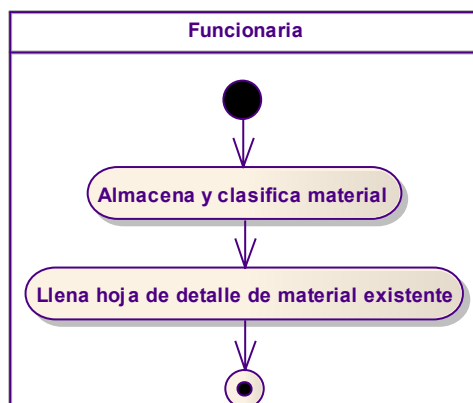


Figura 6.25. Diagrama de Actividades: Almacenar el material: Los Autores.

En el diagrama de actividad de la Figura 6.25 para almacenar el material recibido la Funcionaria separa el material según el tipo, realizado este proceso la Funcionaria procede a llenar la hoja de detalle en la que especifica las existencias del material ingresado.

### Entregar los materiales

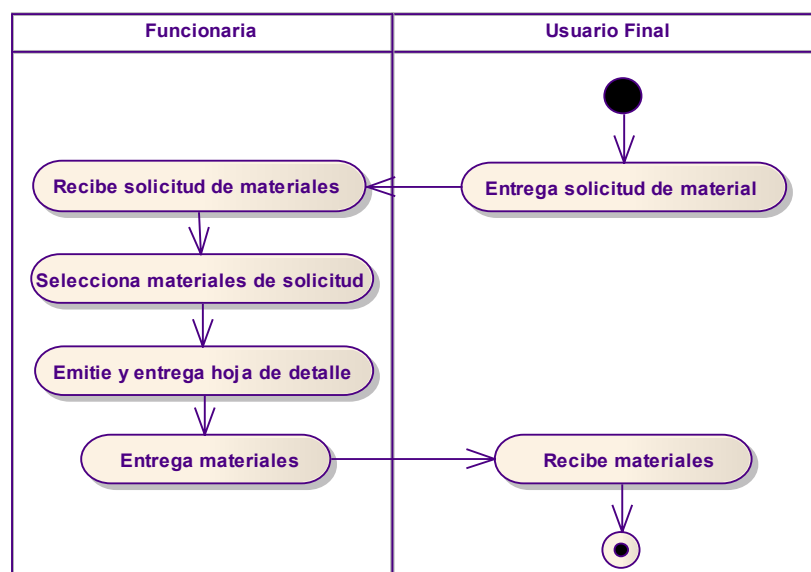


Figura 6.26. Diagrama de Actividades: Entregar los materiales: Los Autores.

En el diagrama de actividad de la Figura 6.26 el Usuario Final procede a entregar la solicitud del material a la Funcionaria, está al recibir la solicitud realiza la selección del material pedido inmediatamente emite y entrega la hoja de detalle y entrega los materiales al Usuario Final.

## 6.3. DIAGRAMAS ESTRUCTURALES

Los diagramas de estructura enfatizan en los elementos que deben existir en el sistema<sup>41</sup>.

### 6.3.1. DIAGRAMA DE COMPONENTES

Diagrama de componentes que permite mostrar una vista física del modelo como también los componentes de software y las relaciones que existen entre ellos<sup>42</sup>.

<sup>41</sup> Lenguaje Unificado de Modelado. [En línea]. <[http://es.wikipedia.org/wiki/Lenguaje\\_Unificado\\_de\\_Modelado](http://es.wikipedia.org/wiki/Lenguaje_Unificado_de_Modelado)>

<sup>42</sup> Romero Moreno, Gesvin. Op. Cit. . p. 28.

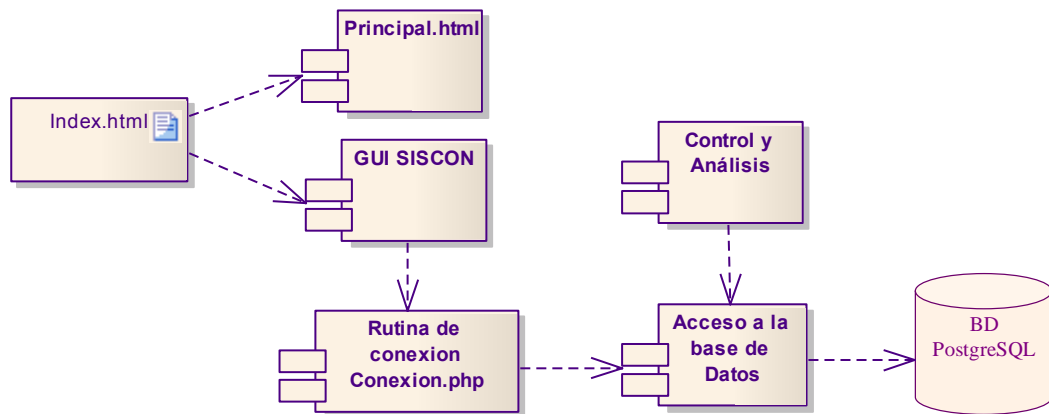


Figura 6.27. Diagramas Estructurales: Diagrama de Componentes: Los Autores.

## CAPÍTULO 7

### 7. PRUEBAS Y CONTROL DE CALIDAD

#### 7.1. PRUEBAS

Una vez terminado el desarrollo del sistema se llevo a efecto un conjunto de pruebas para validar el funcionamiento y consistencia de la aplicación.

Se realizaron los siguientes tipos de prueba: caja negra, de plataforma y de funcionalidad con el usuario; de el resultado de estas se pudo comprobar el adecuado desempeño del mismo.

#### Prueba de funcionalidad

A continuación se presenta las tablas que muestran las acciones que puede realizar el usuario y las respuestas del sistema de control e inventario del Centro de Documentación del Consejo Nacional de las Mujeres.

Las acciones que se describen en las siguientes tablas, están organizadas de acuerdo al orden de aparición del sistema de control e inventario del Centro de Documentación del Consejo Nacional de las Mujeres.

#### Verificar la implementación del acceso al sistema

#	Descripción o contexto de prueba	Entrada o acción del usuario	Reacción o resultado esperado por el sistema	Confirmado
1	Ingresar al sistema con su respectiva cuenta	Nombre de usuario correcto y password correcto	El sistema permite el acceso, identificando correctamente el tipo de usuario y ofreciéndole las opciones correspondientes a su tipo de cuenta	SI
2	Ingresar al sistema con su respectiva cuenta	Nombre de usuario correcto y password incorrecto	El sistema manda un mensaje de error y le manda un link para regresar a la pantalla de ingreso.	SI
3	Ingresar al sistema con su respectiva cuenta	Nombre de usuario correcto y password nulo	El sistema manda un mensaje de error y le manda un link para regresar a la pantalla de ingreso.	SI

4	Ingresar al sistema con su respectiva cuenta	Nombre de usuario incorrecto y password correcto	El sistema manda un mensaje de error y le manda un link para regresar a la pantalla de ingreso.	SI
5	Ingresar al sistema con su respectiva cuenta	Nombre de usuario incorrecto y password incorrecto	El sistema manda un mensaje de error y le manda un link para regresar a la pantalla de ingreso.	SI
6	Ingresar al sistema con su respectiva cuenta	Nombre de usuario incorrecto y password nulo	El sistema manda un mensaje de error y le manda un link para regresar a la pantalla de ingreso.	SI
7	Ingresar al sistema con su respectiva cuenta	Nombre de usuario nulo y password correcto	El sistema manda un mensaje de error y le manda un link para regresar a la pantalla de ingreso.	SI
8	Ingresar al sistema con su respectiva cuenta	Nombre de usuario nulo y password incorrecto	El sistema manda un mensaje de error y le manda un link para regresar a la pantalla de ingreso.	SI
9	Ingresar al sistema con su respectiva cuenta	Nombre de usuario nulo y password nulo.	El sistema manda un mensaje de error y le manda un link para regresar a la pantalla de ingreso.	SI

Tabla 7.1. Prueba de funcionalidad: Verificar la implementación del acceso al sistema: Los autores.

### Verificar la confirmación de cierre de sesión

#	Descripción o contexto de prueba	Entrada o acción del usuario	Reacción o Resultado esperado del Sistema	Confirmado
1	Probar la opción logout	Seleccionar la opción logout en la parte superior del interfaz.	El sistema cierra la sesión y regresa a la pantalla de ingreso de cuenta.	SI

Tabla 7.2. Prueba de funcionalidad: Verificar la confirmación de cierre de sesión: Los Autores.

### Verificar la implementación del menú opciones

#	Descripción o contexto de prueba	Entrada o Acción del Usuario	Reacción o resultado esperado del Sistema	Confirmado
---	----------------------------------	------------------------------	---	------------

1	Probar el menú principal	Seleccionar del menú la opción Catálogo de tablas básicas	El sistema despliega las opciones respectivas del submenú Catálogo de tablas básicas	SI
2	Probar el menú principal	Seleccionar del menú la opción Transacciones	El sistema despliega las opciones respectivas del submenú Transacciones	SI
3	Probar el menú principal	Seleccionar del menú la opción Reportes Estadísticos.	El sistema despliega las opciones respectivas del submenú Reportes Estadísticos.	SI
4	Probar el menú principal	Seleccionar del menú la opción Plan Anual.	El sistema despliega las opciones respectivas del submenú Plan Anual.	SI
5	Probar el menú principal	Seleccionar del menú la opción Ayuda.	El sistema despliega las opciones respectivas del submenú Ayuda.	SI

Tabla 7.3. Prueba de funcionalidad: Verificar la implementación del menú opciones: Los Autores.

### Verificar la lectura con la base de datos

#	Descripción o contexto de prueba	Entrada o acción del usuario	Reacción o resultado esperado por el sistema	Confirmado
1	El sistema intenta conectarse a la base de datos	La base de datos no se encuentra activa o no permite el ingreso	El sistema manda la excepción, para el proceso y despliega un mensaje de error	SI
2	Se cayó o bloqueó la base de datos, o bien el usuario que inició el proceso ha abandonado el sistema	Se perdió la conexión con la base de datos	El sistema manda la excepción, para el proceso y despliega un mensaje de error	SI
3	El sistema intenta leer una tabla de la base de datos	La tabla requerida no se encuentra	El sistema manda la excepción, para el proceso y despliega un mensaje de error	SI
4	El sistema intenta leer una tabla de la base de datos	La tabla requerida no contiene datos	El sistema manda la excepción, para el proceso y despliega un mensaje de error	NO La consulta despliega una tabla vacía

Tabla 7.4. Prueba de funcionalidad: Verificar la lectura con la base de datos: Los Autores.

### Prueba de plataforma

Se realizaron pruebas de la ejecución del sistema en diferentes plataformas siendo las siguientes:

En Windows XP y Windows 2003 Server: Al momento de realizar la prueba del sistema este se ejecutó de manera aceptable y el formato de diseño no tuvo variación alguna.

### Pruebas de Usuario

Se realizaron pruebas de usuario en el sistema con una muestra de 8 usuarios, todos ellos ingresaron, se crearon usuarios para el sistema y probaron todo el sistema en un lapso de varias horas, tiempo en el que el sistema no presentó anomalía alguna y se ejecutó de manera normal.

## 7.2. CONTROL DE CALIDAD

Cuando se evalúa la calidad de un producto de software se debe primeramente establecer ciertos factores y criterios, que de una u otra manera, presente una manifestación de la calidad y el como cuantificarlos.

### Modelo de Calidad

Se pone en práctica el concepto de calidad y describe sus características y relaciones.

### Factores de calidad

Representa la calidad desde el punto de vista usuario, son características que componen la calidad, estos factores se descomponen en criterios de calidad.

Equivalencias:

NRP (número de respuestas positivas).

NT (número total de preguntas de la métrica)

PDC (porcentaje de corrección)

FC (Factor de calidad)

### Integridad

◀ Control de Accesos



¿La interfase con la BDD utiliza los recursos de acceso adecuados a la situación y a las características de perfil de cada usuario?

Si.

¿La interfaz del usuario utiliza control de accesos?

Si.

#### ◀ Facilidad de auditoria

¿Tiene el sistema control de acciones realizados por los usuarios?

No.

¿Tiene el sistema una interfaz que facilite al usuario realizar una auditoria?

No.

#### ◀ Seguridad

¿Existen respaldos del producto software?

Si.

¿Se utiliza los algoritmos de encriptación dentro del sistema?

No.

¿Se encuentra almacenado el código fuente en el servidor principal?

Si.

**NRP = 4**

**NT = 7**

**PDC = (NRP / NT) \* 100**

**PDC = (4 / 7) \* 100**

**PDC = 57.14**

**FC = 0.57**

### **Facilidad de mantenimiento**

#### ◀ Concisión

¿Se ha empleado el reuso de otros elementos del software?

Si.

¿Se han evitado el duplicar líneas de código?

Si.

¿Las funciones del software se encuentran parametrizadas para poder ser utilizadas para otros requerimientos?

Si.

#### ◀ Auto descripción

¿Se ha definido con detalle que pasará con el funcionamiento del software, en caso de que se incurra en un error a la hora alimentar datos, o bien, en caso de un mal funcionamiento interno?

Si.

¿Son claros los mensajes de error?

Si.

¿Es posible saber fácilmente para que sirve cada módulo que comprende el producto de software?

Si.

¿Ayudan los mensajes de error a evitar en reincidir en los errores?

Si.

**NRP = 7**

**NT = 7**

**PDC = (NRP / NT ) \* 100**

**PDC = (7 / 7) \* 100**

**PDC = 100**

**FC = 1**

### **Portabilidad**

#### ◀ Independencia entre sistema y software

¿Es independiente la funcionalidad del sistema de factores relacionados con el software que utiliza?

No.

¿Se ha evitado utilizar características dependientes del sistema operativo o del lenguaje?

Si.

#### ◀ Independencia del hardware

¿Es independiente el diseño del producto de software de características del hardware?

Si.

¿Consume muchos recursos en el momento de la ejecución del software?

CPU. No.

Memoria. No.

Red. No.

¿Es independiente la implementación de producto de software en características del ambiente (Hardware y software)?

Si.

$$\mathbf{NRP = 6}$$

$$\mathbf{NT = 7}$$

$$\mathbf{PDC = (NRP / NT) * 100}$$

$$\mathbf{PDC = (4 / 7) * 100}$$

$$\mathbf{PDC = 85.71}$$

$$\mathbf{FC = 0.85}$$

Equivalencias:

PFC (Ponderación de cada Factor de Calidad)

FC (Factor de calidad)

K (peso ponderado 1 a 5)

PVF (Porcentaje de los valores de factores de calidad con las ponderaciones definidas)

Factor de calidad	Peso (k)
Integridad	5
Facilidad de mantenimiento	3
Portabilidad	5

*Tabla 7.5. Modelo de Calidad: Factor de calidad: Los Autores.*

PVF = integridad + facilidad de mantenimiento + portabilidad

$$\mathbf{PVF = (0.57 * 5) + (1 * 3) + (0.85 * 5)}$$

$$\mathbf{PVF = 2.85 + 3 + 4.25}$$

**PVF = 10.1**

**PVF (óptimo) = 13**

**PVF (obtenido) = 10.1**

El sistema cumple con el 77.69% de Calidad de Software de acuerdo a los factores y criterios seleccionados.

### **Lista de chequeo de estándares de presentación y funcionalidad de la aplicación para formas**

<b>Forma:</b> Funcionaria			
<b>Responsable:</b> Diego Rueda			
<b>Descripción:</b> Formulario para el ingreso de información de funcionarias.			
<b>Revisión de estándares de presentación</b>			
<b>Actividad</b>	<b>Sí</b>	<b>No</b>	<b>Información adicional</b>
¿Están claramente definidos los bloques de información (Frames)?	x		
¿Tiene los encabezados de título y nombre de aplicación correctos?	x		
¿Las etiquetas de los campos son claras y representativas?	x		
¿Los campos de solamente despliegue están claramente identificados?	x		
¿Tiene los colores estándar?	x		
¿Los campos fecha tienen el formato DD-MM-AAAA y se puede ingresar los datos como Ej: 12ago2010?		x	El formulario no tiene campo para ingreso de fecha
¿Los campos están alineados en forma correcta?	x		
¿Las barras de Scroll son blancas?	x		
¿La forma realiza la función que se necesita?	x		
¿Es rápido y fácil el manejo de la forma?	x		
Los campos que hacen referencia a datos de otras tablas ¿tienen cada uno su lista de valores?	x		
¿Las listas de valores son lentas para recuperar la información?		x	
¿El tiempo de respuesta es adecuado?	x		
¿El orden de navegación de los campos es el correcto?	x		
¿Los mensajes graves son manejados adecuadamente?	x		

¿Si el reporte requiere mucho tiempo, esto le es notificado al usuario?		x	
¿Está la forma documentada?	x		

Tabla 7.6. Lista de Chequeo: Funcionaria: Los Autores.

**FUNCIONARIA**

CEDULA:

NOMBRES:

APELLIDOS:

DIRECCION:

TELEFONO:

CELULAR:

USUARIO:

PASSWORD:

PRIVILEGIO:

[Ver Funcionarias](#)

Figura 7.1. Lista de Chequeo: Funcionaria: Los Autores.

<b>Forma:</b> Productos			
<b>Responsable:</b> Diego Rueda			
<b>Descripción:</b> Formulario para el ingreso de información de los productos ha ser usados en el Centro de Documentación y Difusión.			
<b>Revisión de estándares de presentación</b>			
Actividad	Sí	No	Información adicional
¿Están claramente definidos los bloques de información (Frames)?	x		
¿Tiene los encabezados de título y nombre de aplicación correctos?	x		
¿Las etiquetas de los campos son claras y representativas?	x		
¿Los campos de solamente despliegue están claramente identificados?	x		
¿Tiene los colores estándar?	x		
¿Los campos fecha tienen el formato DD-MM-AAAA y se puede ingresar los datos como Ej: 12ago2010?		x	El formulario no tiene campo para ingreso de fecha
¿Los campos están alineados en forma correcta?	x		
¿Las barras de Scroll son blancas?	x		

¿La forma realiza la función que se necesita?	x		
¿Es rápido y fácil el manejo de la forma?	x		
Los campos que hacen referencia a datos de otras tablas ¿tienen cada uno su lista de valores?	x		
¿Las listas de valores son lentas para recuperar la información?		x	
¿El tiempo de respuesta es adecuado?	x		
¿El orden de navegación de los campos es el correcto?	x		
¿Los mensajes graves son manejados adecuadamente?	x		
¿Si el reporte requiere mucho tiempo, esto le es notificado al usuario?		x	
¿Está la forma documentada?	x		

Tabla 7.7. Lista de Chequeo: Productos: Los Autores.

**PRODUCTOS**

CODIGO:

DESCRIPCION:

STOCK MAXIMO:

STOCK MINIMO :

EXISTENCIA:

TIPO DE UNIDAD:

TIPO DE PRODUCTO:

[Ver Productos](#)

Figura 7.2. Lista de Chequeo: Productos: Los Autores.

**Forma:** Egresos

**Responsable:** Diego Rueda

**Descripción:** Formulario para el ingreso de información de los productos ha ser usados en el Centro de Documentación y Difusión.

**Revisión de estándares de presentación**

Actividad	Sí	No	Información adicional
¿Están claramente definidos los bloques de información (Frames)?	x		
¿Tiene los encabezados de título y nombre de aplicación correctos?	x		
¿Las etiquetas de los campos son claras y	x		

representativas?			
¿Los campos de solamente despliegue están claramente identificados?	x		
¿Tiene los colores estándar?	x		
¿Los campos fecha tienen el formato DD-MM-AAAA y se puede ingresar los datos como Ej: 12ago2010?	x		
¿Los campos están alineados en forma correcta?	x		
¿Las barras de Scroll son blancas?	x		
¿La forma realiza la función que se necesita?	x		
¿Es rápido y fácil el manejo de la forma?	x		
Los campos que hacen referencia a datos de otras tablas ¿tienen cada uno su lista de valores?	x		
¿Las listas de valores son lentas para recuperar la información?		x	
¿El tiempo de respuesta es adecuado?	x		
¿El orden de navegación de los campos es el correcto?	x		
¿Los mensajes graves son manejados adecuadamente?	x		
¿Si el reporte requiere mucho tiempo, esto le es notificado al usuario?		x	
¿Está la forma documentada?	x		

Tabla 7.8. Lista de Chequeo: Egresos: Los Autores.

**CABECERA**

AÑO  
TRANSACTION: **2011**

RESPONSABLE: 1715345565: ALVAREZ CARLOS ▼

USUARIO FINAL: Ministerio de Justicia ▼

UBICACION  
GEOGRAFICA: ECUADOR ▼

ORGANIZACION: Consejo Nacional de las Mujeres

FECHA  
TRANSACTION: **2011-7-1**

DETALLE DE  
TRANSACTION:

TIPO DE  
DOCUMENTO: HOJA DE RUTA ▼

TIPO DE  
TRANSACTION: CHARLAS O CONFERENCIA ▼

[Ver Egresos](#)

Figura 7.3. Lista de Chequeo: Cabecera: Los Autores.

## Lista de chequeo de estándares de presentación y funcionalidad de la aplicación para reportes

<b>Reporte:</b> Movimiento por Destinatario			
<b>Responsable:</b> Diego Rueda			
<b>Descripción:</b> Reporte para el despliegue de datos según movimiento por destinatario.			
<b>Revisión de estándares de presentación</b>			
Actividad	Sí	No	Información adicional
¿El reporte tiene el nombre del sistema correcto?	x		
¿El reporte tiene los encabezados de título y nombre de reporte correctos?	x		
¿El reporte tiene la fecha de generación?	x		
¿El reporte tiene el número de página y el total de páginas?	x		
¿Están claramente definidos los bloques de información?	x		
¿Las etiquetas de los campos son claras y representativas?	x		
¿El reporte tiene el logo y los colores estándares de la institución?	x		
¿Los campos fecha tienen el formato DD-MON-YYYY?	x		
¿Los campos están alineados en forma correcta?	x		
¿El reporte tiene enumeradas las filas?	x		
¿El reporte tiene el visto bueno del usuario?	x		

Tabla 7.9. Lista de Chequeo: Movimiento por Destinatario: Los Autores.

<b>Reporte:</b> Egresos Totales			
<b>Responsable:</b> Diego Rueda			
<b>Descripción:</b> Reporte para el despliegue de datos para los Egresos Totales.			
<b>Revisión de estándares de presentación</b>			
Actividad	Sí	No	Información adicional
¿El reporte tiene el nombre del sistema correcto?	x		
¿El reporte tiene los encabezados de título y nombre de reporte correctos?	x		
¿El reporte tiene la fecha de generación?	x		
¿El reporte tiene el número de página y el total de páginas?	x		
¿Están claramente definidos los bloques de información?	x		
¿Las etiquetas de los campos son claras y representativas?	x		
¿El reporte tiene el logo y los colores	x		



estándares de la institución?			
¿Los campos fecha tienen el formato DD-MON-YYYY?	x		
¿Los campos están alineados en forma correcta?	x		
¿El reporte tiene enumeradas las filas?	x		
¿El reporte tiene el visto bueno del usuario?	x		

Tabla 7.10. Lista de Chequeo: Egresos Totales: Los Autores.

### Lista de chequeo de estándares de tablas

<b>Tabla:</b> Funcionaria <b>Responsable:</b> Diego Rueda <b>Descripción:</b> Tabla para el almacenamiento de información referente a datos personales de Funcionarias.			
<b>Revisión de estándares de presentación</b>			
Actividad	Sí	No	Información adicional
¿El nombre de la tabla es correcto según los estándares de la institución?	x		
¿Tiene las descripciones de la columna en la base de datos?		x	Tiene la descripción para los campos mas relevantes como código
¿Tiene las llaves e índices adecuados?	X		
¿La tabla ha sido recreada teniendo en cuenta su uso?	x		

Tabla 7.11. Lista de Chequeo: Funcionaria: Los Autores.

<b>Tabla:</b> Producto <b>Responsable:</b> Diego Rueda <b>Descripción:</b> Tabla para el almacenamiento de información referente a datos de productos.			
<b>Revisión de estándares de presentación</b>			
Actividad	Sí	No	Información adicional
¿El nombre de la tabla es correcto según los estándares de la institución?	x		
¿Tiene las descripciones de la columna en la base de datos?		x	Tiene la descripción para los campos mas relevantes como código
¿Tiene las llaves e índices adecuados?	X		
¿La tabla ha sido recreada teniendo en cuenta su uso?	x		

Tabla 7.12. Lista de Chequeo: Producto: Los Autores.

## CAPÍTULO 8

### CONCLUSIONES

- El software libre siendo una herramienta de libre distribución y la libertad de usar los programas con cualquier propósito fue un gran aliado al momento de desarrollar y programar el sistema para el control e inventario de los materiales de donación y difusión.
- Los reportes provistos mediante el sistema a las funcionarias redujo su tiempo de trabajo al presentarles la información de una manera adecuada, entendible y consolidada además con la facilidad de exportarlos a varios formatos.
- El sistema desarrollado para el control e inventario de los materiales de donación y difusión hace que los ingresos y egresos sean almacenados en una forma más organizada y estructurada y además las consultas y reportes obtenidos son más claros y entendibles.
- La interfaz de usuario del sistema hace que la funcionaria mediante su menú de opciones y pantallas de ingreso o consulta use la aplicación de una manera más amigable y ágil.
- Se ha aplicado para el sistema los conocimientos obtenidos en las materias dadas por la Universidad Politécnica Salesiana y además de conocimientos obtenidos de varias fuentes bibliográficas y links de Internet., lo cual ha aportado en forma fundamental a los conocimientos de los integrantes del grupo y en el desarrollo del sistema.
- Además se ha aplicado para el sistema los conocimientos obtenidos de varias fuentes bibliográficas y links de Internet sobre el uso de la herramienta de base de datos postgresQL, lo cual ha aportado en forma fundamental a los conocimientos de los integrantes del grupo y en el desarrollo del sistema.

## RECOMENDACIONES

- Se debería dar mayor prioridad al uso y aprendizaje de herramientas de software libre para el desarrollo de aplicaciones.
- Al momento de presentar reportes estos deben estar bien detallados, organizados y con los datos necesarios y específicos solicitados por el usuario o usuaria para hacer su labor más eficaz, ágil y concisa.
- Se debe estructurar desde el principio una óptima base de datos para que el sistema que se desarrolle sea seguro y organizado, y las consultas que se hagan tengan una respuesta ágil y exacta.
- Los sistemas deben ser desarrollados con interfaces de usuario sujeto a un modelo con características amigables, fácil de manipular e intuitivo al momento de usar la aplicación por el usuario.
- El Internet es una herramienta muy fundamental para adquirir conocimientos y autoeducación, además la información obtenida en las diferentes cátedras dadas en la Universidad Politécnica Salesiana son de bastante ayuda al momento de realizar investigaciones y al momento de desarrollar aplicaciones informáticas.

## REFERENCIAS BIBLIOGRÁFICAS

- De la Cruz Villar, Joel. *PHP Y MySQL*. Lima: Grupo Editorial Megabyte, 2006.
- Romero Moreno, Gesvin. *UML con Rational Rose*. Lima: Grupo Editorial Megabyte, 2004.
- Jacobson, Ivar – Booch, Grandy. *EL PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE*. Madrid: Pearson Educación, 2000.
- Normas para trabajos Académicos. [en línea].  
<http://www.slideshare.net/Igbernal8606/normas-icontec-trabajos-acadmicos>  
[Citado el 6 de mayo de 2010].
- Manual para presentación de trabajos académicos. [en línea].  
<http://ebookbrowse.com/manual-para-elaboracion-y-presentacion-de-trabajos-academicos-pdf-d31690530>  
[Citado el 10 de mayo de 2010].
- El arte de escribir código mantenible. [en línea].  
<http://www.infosum.net/es/programming/the-art-of-writing-maintainable-code.html>  
[Citado el 18 de mayo de 2010].
- Diccionario básico de informática. [en línea].  
<http://usuarios.multimania.es/Resve/diccioninform.htm>  
[Citado el 28 de Junio de 2010].
- Definición de ASCII. [en línea].  
<http://www.pergaminovirtual.com/definicion/ASCII.html>  
[Citado el 10 de Julio de 2010].
- Sesiones en Php. [en línea].  
<http://www.programacionweb.net/articulos/articulo/?num=377>  
[Citado el 5 de Agosto de 2010].
- JavaScript. [en línea].  
<http://www.librosweb.es/javascript>  
[Citado el 28 de Septiembre de 2010].
- XHTML. [en línea].  
[http://www.mati.unam.mx/index.php?option=com\\_content&task=view&id=205&Itemid=51](http://www.mati.unam.mx/index.php?option=com_content&task=view&id=205&Itemid=51)  
[Citado el 17 de Octubre de 2010].

- Widget. [en línea].  
<http://ayuda.codeeta.com/que-es-un-widget/>  
[Citado el 20 de Octubre de 2010].
- Estructuras de Control. [En línea].  
[http://www.librosweb.es/javascript/capitulo3/estructuras\\_de\\_control\\_de\\_flujo.html](http://www.librosweb.es/javascript/capitulo3/estructuras_de_control_de_flujo.html)  
[Citado el 30 de Octubre de 2010].
- Estructuras de Control. [En línea].  
[http://www.librosweb.es/javascript/capitulo3/estructuras\\_de\\_control\\_de\\_flujo.html](http://www.librosweb.es/javascript/capitulo3/estructuras_de_control_de_flujo.html) [Citado el 5 de Noviembre de 2010].
- Funciones y Propiedades. [En línea].  
[http://www.librosweb.es/javascript/capitulo3/funciones\\_y\\_propiedades\\_basicas\\_de\\_javascript.html](http://www.librosweb.es/javascript/capitulo3/funciones_y_propiedades_basicas_de_javascript.html)  
[Citado el 8 Noviembre de 2010].
- UML. [En línea].  
[http://es.wikipedia.org/wiki/Lenguaje\\_Unificado\\_de\\_Modelado](http://es.wikipedia.org/wiki/Lenguaje_Unificado_de_Modelado)  
[Citado el 16 Noviembre de 2010].
- Especificación casos de uso. [En línea].  
[http://www.cyta.com.ar/cta/proyectos/pi\\_sitio\\_htm/material/3\\_plantilla\\_especificacion\\_caso\\_Uso.doc](http://www.cyta.com.ar/cta/proyectos/pi_sitio_htm/material/3_plantilla_especificacion_caso_Uso.doc)  
[Citado el 20 Noviembre de 2010].
- Definición de ofimática.  
<http://es.wikipedia.org/wiki/Ofim%C3%A1tica>  
[Citado el 22 Noviembre de 2010].
- Manual de Usuario de PostgreSQL. [En línea].  
<http://postgresql.uci.cu/system/files/Manual%20del%20usuario%20de%20PostgreSQL.pdf>  
[Citado el 28 Noviembre de 2010].
- Instalación Grafica de PostgreSQL. [En línea].  
<http://www.postgresql-es.org/node/361>  
[Citado el 30 Noviembre de 2010].
- Introducción a PostgreSQL-Configuración. [En línea].  
<http://www.linux-es.org/node/660>  
[Citado el 5 Diciembre de 2010].
- Descargar PostgreSQL. [En línea].  
<http://www.enterprisedb.com/products/pgdownload.do>  
[Citado el 8 Diciembre de 2010].

- Lenguaje Unificado de Modelado. [En línea].  
[http://es.wikipedia.org/wiki/Lenguaje\\_Unificado\\_de\\_Modelado](http://es.wikipedia.org/wiki/Lenguaje_Unificado_de_Modelado)  
[Citado el 10 Diciembre de 2010].
- Calidad de Software. [En línea].  
[http://gidis.ing.unlpam.edu.ar/downloads/pdfs/Calidad\\_software.PDF](http://gidis.ing.unlpam.edu.ar/downloads/pdfs/Calidad_software.PDF)  
[Citado el 15 Diciembre de 2010].
- Arquitectura de Diagramas. [En línea].  
<http://www.scribd.com/Documento-de-Arquitectura-Diagramas/d/10074807>  
[Citado el 20 Diciembre de 2010].

**ANEXOS**

# CONTENIDO

I

**pág.**

LISTA DE ILUSTRACIONES

II

## **MANUAL DE INSTALACIÓN Y CONFIGURACIÓN DE POSTGRESQL**

1

1. Requisitos

1

2. Instalación de la base de datos

1

3. Instalación del servidor Apache

2



## LISTA DE ILUSTRACIONES

	pág.
<b>Figura 1.</b> Pantalla principal de PostgreSQL	1
<b>Figura 2.</b> Pantalla cambiar contraseña de PostgreSQL	2
<b>Figura 3.</b> Instalador Xampp	2
<b>Figura 4.</b> XAMPP Control Panel Application	3
<b>Figura 5.</b> Información Php	3
<b>Figura 6.</b> Directorio Xampp	4
<b>Figura 7.</b> Archivo php.ini	4

# MANUAL DE INSTALACIÓN Y CONFIGURACIÓN DE POSTGRESQL

A continuación se presenta todas las instrucciones para realizar la configuración de postgresQL.

## Requisitos:

- Base de Datos PostgreSQL 8.4
- Servidor Web APACHE
- Browser (Internet Explorer 8)

## Instalación:

Configuración del Cliente y Usuario de la Base de Datos.

1. Se ingresa a PostgreSQL 8.4 luego se procede a digitar el usuario y la contraseña.

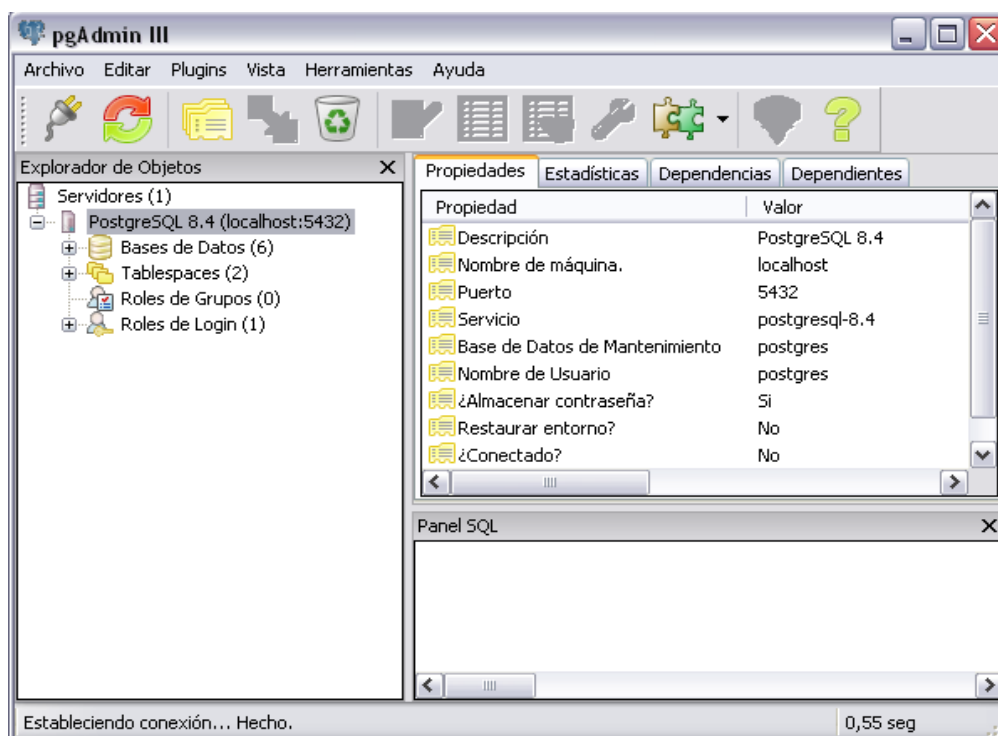
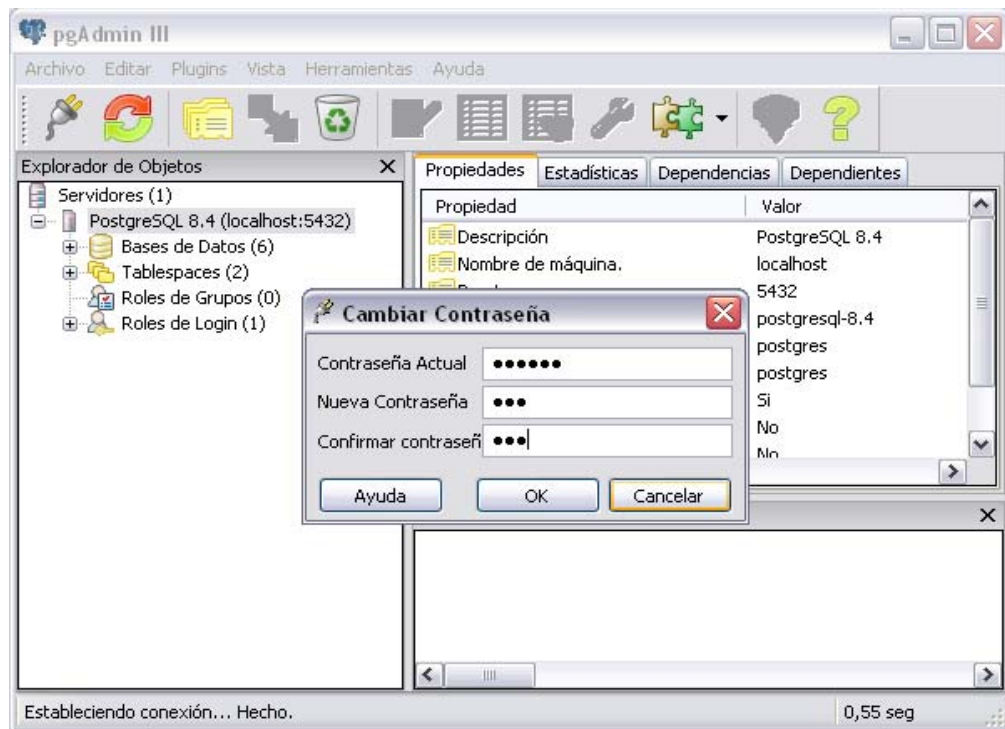


Figura1. Pantalla principal de PostgreSQL: Los Autores.

2. En el menú principal ubicamos a la opción Archivo y se procede a elegir Cambiar contraseña, a la que se asignará es "123".



*Figura2. Pantalla cambiar contraseña de PostgreSQL: Los Autores.*

3. Se crea una nueva base de datos y se procede a importar el script de la base de datos.

## **Instalación del Servidor Apache**

1. Para instalar XAMPP sólo se requiere descargar XAMPP en: <http://www.apachefriends.org/en/xampp.html> y extraer XAMPP. XAMPP es una forma fácil de instalar la distribución Apache que contiene MySQL, PHP y Perl.



*Figura3. Instalador xampp: Los Autores.*

2. Una vez instalado se verifica que la aplicación este ejecutándose correctamente en XAMPP Control Panel Application. Es decir que Apache este “running”.

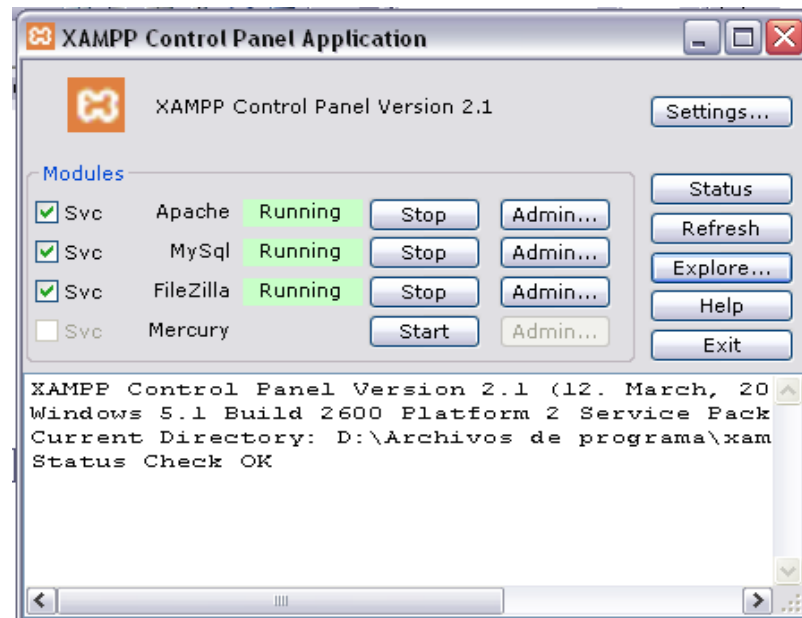


Figura4. XAMPP Control Panel Application: Los Autores.

3. Se verifica qué ini esta usando xampp escribiendo en la barra de direcciones localhost/xampp/phpinfo.php en este caso es el que se encuentra en D:\Archivos de programa\xampp\apache\bin\php.ini

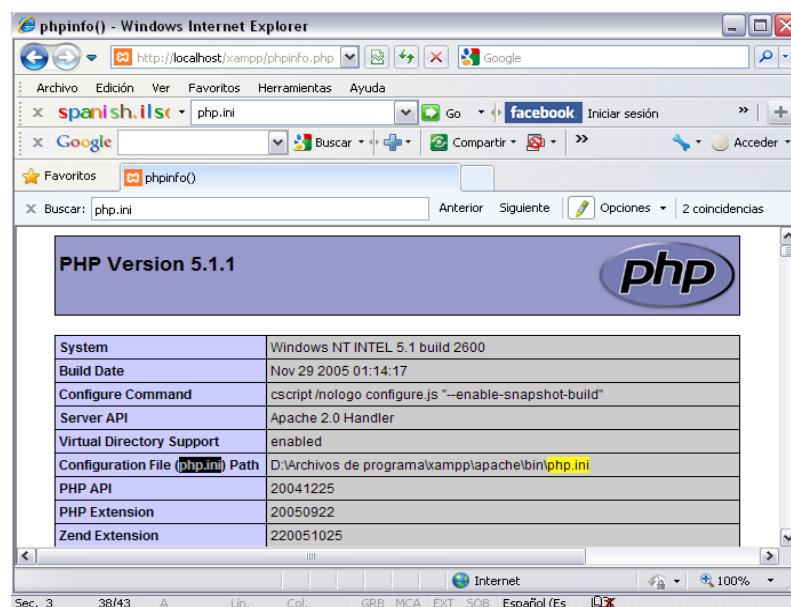


Figura5. Información Php: Los Autores.

4. Se realiza algunas modificaciones en archivos, debido a que suelen aparecer algunos problemas con la integración de php y postgres.

Se debe hacer la modificación de un archivo para que apache, php y postgres funcionen conjuntamente. Pero resulta que xampp posee tres de estos archivos en:

D:\Archivos de programa\xampp\php\php.ini

D:\Archivos de programa\xampp\php\php5.ini

D:\Archivos de programa\xampp\apache\bin\php.ini

En este caso al que se le hace la modificación es al tercer archivo ya que este es el que esta usando xampp.

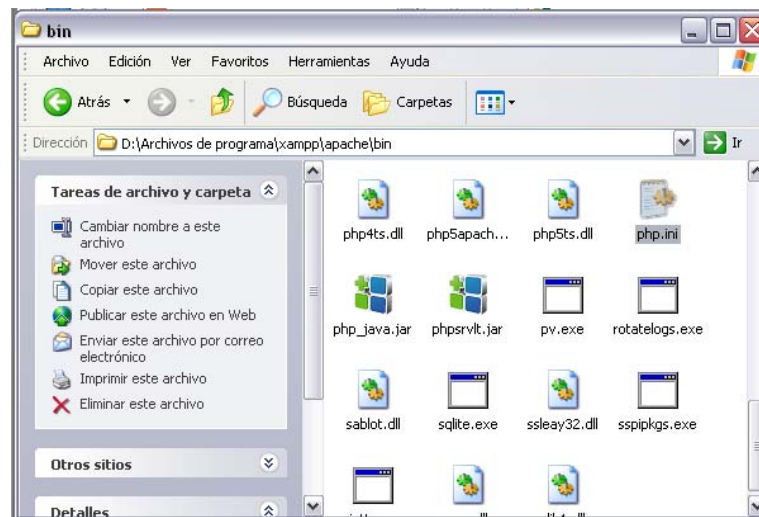


Figura6. Directorio Xampp: Los Autores.

5. Se abre el archivo php.ini y en la línea `;extension=php_pgsql.dll` se procede a quitar el ( ; ) del inicio de esta sentencia y se guarda la modificación realizada en el archivo.

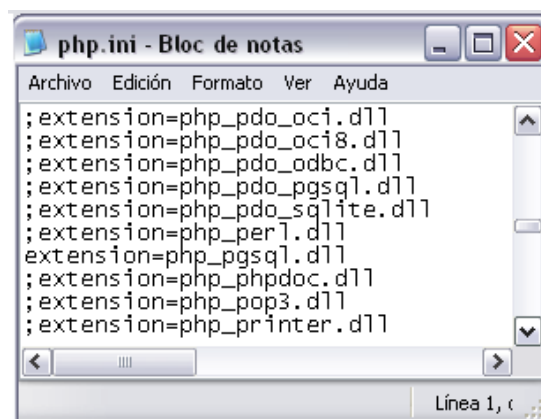


Figura7. Archivo php.ini: Los Autores.

**CONTENIDO**

	<b>pág.</b>
LISTA DE ILUSTRACIONES	II
<b>MANUAL DE USUARIO</b>	<b>1</b>
Introducción	1
1. Ingreso a la aplicación	1
2. Pantalla de inicio o home	2
3. Ingreso de datos en los formularios	4
4. Modificación de registros	5
5. Despliegue de registros	5
6. Formulario de encuesta virtual	6
7. Ingreso de nuevos productos	6
8. Generar nuevo ingreso	7
9. Generar nuevo egreso	8
10. Reportes	9
11. Manual de Usuario	10

**LISTA DE ILUSTRACIONES**

	<b>pág.</b>
<b>Figura 1.</b> Pantalla de Ingreso a la Aplicación	1
<b>Figura 2.</b> Pantalla de Inicio o Home	2
<b>Figura 3.</b> Pantalla de Navegación o Menú	2
<b>Figura 4.</b> Pantalla de Ingreso a la Aplicación	3
<b>Figura 5.</b> Pantalla de Ingreso de datos	4
<b>Figura 6.</b> Pantalla Almacenar datos	4
<b>Figura 7.</b> Pantalla de Modificación de registros	5
<b>Figura 8.</b> Pantalla de Despliegue de registros	5
<b>Figura 9.</b> Pantalla del Formulario de encuesta virtual	6
<b>Figura 10.</b> Pantalla de Ingreso de nuevos productos	6
<b>Figura 11.</b> Pantalla de Generar nuevo ingreso	7
<b>Figura 12.</b> Pantalla Ver ingresos	8
<b>Figura 13.</b> Pantalla Generar nuevo egreso	8
<b>Figura 14.</b> Pantalla Ver egresos	9
<b>Figura 15.</b> Pantalla Reportes	10
<b>Figura 16.</b> Pantalla Manual de Usuario	10

## MANUAL DE USUARIO

### INTRODUCCIÓN

La herramienta que se explica a continuación fue creada con el fin de controlar e inventariar los materiales con los cuáles se trabajan en el Centro de Difusión y Documentación del Consejo Nacional de las Mujeres. Mediante el siguiente sistema la funcionaria podrá controlar la información con la cual se trabaja para realizar los ingresos, los egresos, la evaluación de atención al usuario y una planificación anual del Centro de Difusión y Documentación.

El manual que se presenta a continuación guiará a la funcionaria paso a paso por todo el sistema y mostrará mediante pantallas ilustrativas el funcionamiento de la misma.

### 1. Ingreso a la aplicación

Para acceder al sistema, realice los siguientes pasos:

- 1.1. Ingrese la siguiente dirección en su navegador de Internet:  
<http://localhost/siscon>, (para ingresar al sistema de control e inventario del CONAMU, se debe ver cual será la IP del servidor donde vaya a estar instalado el sistema)
- 1.2. Se mostrará una pantalla en la que deberá ingresar su nombre de usuario y contraseña.



CONAMU  
CONSEJO NACIONAL DE LAS MUJERES  
PRESIDENCIA DE LA REPÚBLICA

**Sistema De Control  
e Inventario  
Centro de Documentación  
y Difusión**

Usuario:

Contraseña:

Figura 1. Pantalla de Ingreso a la Aplicación: Los Autores.



- 1.3. Una vez ingresados sus datos, presione **Aceptar**. Se abrirá la pantalla de inicio o home.

## 2. Pantalla de inicio o home



Figura 2. Pantalla de Inicio o Home: Los Autores.

De aquí en adelante, las distintas pantallas a las que ingrese conforme navegue a través del sistema, se dividirán en dos partes:

- Barra de navegación o Menú.
- Área de Despliegue.

### 2.1. Barra de Navegación o Menú



Figura 3. Pantalla de Navegación o Menú: Los Autores.

Muestra el menú principal desde el cual usted podrá ingresar a las distintas áreas que ofrece el sistema. Cada área consta de formularios para ser llenados de acuerdo a la elección del usuario. Para acceder a ellas, simplemente ubique el cursor sobre la opción que desee visualizar y haga clic sobre ella. Se desplegará automáticamente un formulario en el área de despliegue. En la parte superior del menú, se encuentra un link para regresar a la pantalla principal o **home**.

## 2.2. Área de Despliegue

*Figura 4. Pantalla de Ingreso a la Aplicación: Los Autores.*

Aquí se desplegarán las pantallas según la opción que se haya elegido, predeterminado está un formulario para el ingreso de datos, además cada pantalla tiene en la parte inferior un link que permite visualizar los registros existentes en cada tabla

### 3. Ingreso de datos en los formularios

Figura 5. Pantalla de Ingreso de datos: Los Autores.

Al escoger la opción que se desea del menú, se desplegará un formulario donde se debe llenar los campos requeridos, muchos de los formularios tendrán campos tipo combo box, en el cuál bastará con elegir el dato que se necesita, también hay campos donde permitirá el ingreso únicamente de números como por ejemplo en los campos para cédula, evitando así que se ingresen datos incorrectos.

Una vez ingresados los datos, hacer clic en el botón **Guardar**, los datos serán grabados en sus respectivas tablas, automáticamente se desplegará una pantalla con los datos existentes en la respectiva tabla.

CEDULA	NOMBRES FUNCIONARIA
1234567890	aaa aaa
1111111111	Ana Lucia Guanochangea
1716957418	Diego Fernando Rueda Macas
4545454545	J. G.
2222222222	Luchita G:G

Figura 6. Pantalla Almacenar datos: Los Autores.

#### 4. Modificación de registros

Si se observa los datos de las primera columna están con un link en el cual al dar un clic va hacia a una pantalla de actualización de datos donde se puede ingresar la información que se desea cambiar, una vez llenados los campos que se desea modificar se da un clic en el botón **Guardar Cambios**, y automáticamente los datos serán actualizados y se regresa a la pantalla donde están desplegados los registros existentes.

Figura 7. Pantalla de Modificación de registros: Los Autores.

#### 5. Despliegue de registros

En la pantalla del formulario de ingreso, se nota que en la parte inferior de cada formulario hay un link que dice **Ver [tabla]** si se da un clic aquí se desplegará la pantalla de registros existentes, si se quiere volver a ingresar un nuevo registro, se debe dar clic en el botón **Volver**.

COMISION DE TRANSICION SISTEMA DE CONTROL E INVENTARIO	
PRODUCTOS	
CODIGO	DESCRIPCION
209	CD COMPILACION DE INSTRUMENTOS INTERNACIONALES
210	CD CONSEGUIR SALUD PARA TODO EL MUNDO/PRB
211	CD ECUADOR BEIJING +10
212	CD CUÁAS DE AUDIO DE DERECHOS DE LAS MUJERES
213	VIDEO DE LA ESCUELA QUE QUEREMOS
215	VIDEO NADA JUSTIFICA LA VIOLENCIA
216	CD DILE NO AL ACOSO SEXUAL
217	CUADRIPTICO MUJER LA CONSTITUCION TE DA DERECHOS
221	VOLANTES PAE
222	VOLANTE CONSEJOS UTILES EN CASO DE VIOLENCIA
223	TRIPTICO PRO MUJERES
226	PROTOCOLO FACULTATIVO CEDAW
227	PROGRAMA Y ACCION DE VIENA
228	LMGYAI
229	MANUAL DE PROCEDIMIENTOS LEY 103
230	CON MUJERES #1
231	CON MUJERES #2
232	MUJERES Y HOMBRES DEL ECUADOR EN CIFRAS II

Figura 8. Pantalla de Despliegue de registros: Los Autores.

## 6. Formulario de Encuesta Virtual

Para acceder a esta pantalla hay que ingresar primeramente con el usuario respectivo para presentar y acceder a esta opción. Una vez que se ingresa al sistema, posesionarse sobre la pantalla de menú y dar clic en el submenú **Ayuda**, se despliega tres opciones, **escoger** la opción **Encuesta Virtual CENDOC**. Automáticamente se desplegará una pantalla con el formulario de la encuesta que debe ser llenado por cada usuario del Centro de Documentación y Difusión.

EVALUACION CENTRO DE DOCUMENTACION

CEDULA:

NOMBRES:

APELLIDOS:

SEXO: Femenino

INSTITUCION:

EMAIL:

TEMA DE CONSULTA:

TELEFONO:

FECHA CONSULTA: 2011-3-17

TIPO DE USUARIO:

TIPO DE SERVICIO:

[Ver Evaluaciones](#)

Figura 9. Pantalla del Formulario de encuesta virtual: Los Autores.

Una vez llenado los campos requeridos dar clic en el botón **Guardar**, para ver el detalle de las evaluaciones realizadas dar clic en el link **Ver Evaluaciones**.

## 7. Ingreso de Nuevos Productos

PRODUCTOS

CODIGO:

DESCRIPCION:

STOCK MAXIMO:

STOCK MINIMO:

EXISTENCIA:

TIPO DE UNIDAD: Resma

TIPO DE PRODUCTO: Revistas

[Ver Productos](#)

Home

- Catalogos de tablas basicas
  - Usuarios
    - Funcionaria
    - Tipo de Privilegios
    - Caracteristicas
  - Productos
    - Formato de Tipo de Producto
    - Productos
    - Tipo de Unidad
    - Tipo de Producto
    - Proveedor
  - Transaccion
    - Evaluacion Atencion
    - Tipo de Usuario
    - Tipo de Servicio
  - Transacciones
    - Ingresos
      - Generar nuevo Ingreso
    - Egresos
    - Reporte Estadistico de

Figura 10. Pantalla de Ingreso de nuevos productos: Los Autores.

Para ingresar un nuevo producto es de igual manera que los anteriores formularios, buscar el submenú **Productos** e ir a la opción **Productos** y dar un clic, se desplegará un formulario en el cual se deberá llenar los respectivos campos, después dar clic en el botón **Guardar**, para revisar los productos almacenados hay q ir a la parte inferior y dar clic en el link **Ver Productos**.

## 8. Generar nuevo ingreso

Figura 11. Pantalla de Generar nuevo ingreso: Los Autores.

Para generar un nuevo ingreso, primero hay que ir al submenú **Transacciones**, luego a **Ingresos** y dar clic en la opción **Generar nuevo ingreso**, se desplegará un formulario de Cabecera, llenar los respectivos datos y dar clic en el botón **Guardar**, automáticamente los datos se almacenarán y a continuación se desplegará una pantalla con un formulario para el ingreso del detalle de los productos que son ingresados sea por compra, devolución, donación, etc. Para revisar los registros de ingresos ya realizados dar clic en el link **Ver Ingresos**, que se encuentra en la parte inferior del formulario.

**CABECERA**

Codigo Transaccion: 11

Año Transaccion: 2011

Proveedor: EDITORIAL LATINA

Organizacion: EDITORIAL LATINA

Detalle: ddd

Fecha: 2011-03-17

Tipo de Documento: HOJA DE RUTA

Tipo de Transaccion: CHARLAS O CONFERENCIA

**PRODUCTOS**

Producto: ACCION POSITIVA/MEMORIA SEMIN REGIONAL MECANISMOS

Descripción: Libros

Cantidad: 3

Añadir Productos

CODIGO	DESCRIPCION	CANTIDAD
264	ACCION POSITIVA/MEMORIA SEMIN REGIONAL MECANISMOS	3

Finalizar

Figura 12. Pantalla Ver ingresos: Los Autores.

En esta pantalla ingresar los datos requeridos y dar un clic en el botón **Añadir Productos**, se irán añadiendo cuantos productos se quieran ingresar para terminar dar clic en el botón **Finalizar**.

## 9. Generar nuevo egreso

**CABECERA**

AÑO TRANSACCION: 2011

RESPONSABLE: 1715345565: ALVAREZ CARLOS

USUARIO: Ministerio de Justicia

FINAL: Ministerio de Justicia

UBICACION: ECUADOR

GEOGRAFICA: ECUADOR

ORGANIZACION: Consejo Nacional de las Mujeres

FECHA TRANSACCION: 2011-3-17

DETALLE DE TRANSACCION:

TIPO DE DOCUMENTO: HOJA DE RUTA

TIPO DE TRANSACCION: CHARLAS O CONFERENCIA

Guardar Cancelar

[Ver Egresos](#)

Figura 13. Pantalla Generar nuevo egreso: Los Autores.

Para generar un nuevo egreso, primero hay que ir al submenú **Transacciones**, luego a **Egresos** y dar clic en la opción **Generar nuevo egreso**, se desplegará un formulario de Cabecera, llenar los respectivos datos y dar clic en el botón **Guardar**, automáticamente los datos se almacenarán y a continuación se

desplegará una pantalla con un formulario para el ingreso del detalle de los productos que son egresados sea por donación o difusión. Para revisar los registros de egresos ya realizados dar clic en el link **Ver Egresos**, que se encuentra en la parte inferior del formulario.

**CABECERA**

<b>Codigo Transaccion:</b>	29	<b>Fecha:</b>	2011-03-17
<b>Año Transaccion:</b>	2011	<b>Tipo de Documento:</b>	HOJA DE RUTA
<b>Responsable:</b>	ALVAREZ CARLOS	<b>Tipo de Transaccion:</b>	CHARLAS O CONFERENCIA
<b>Organizacion:</b>	Consejo Nacional de las Mujeres	<b>Detalle:</b>	
<b>Usuario Final:</b>	Ministerio de Justicia		

**PRODUCTOS**

**Producto:**

**Descripción:**

**Cantidad:**

CODIGO	DESCRIPCION	CANTIDAD
244	BASE POLITICA FORO DE LA MUJER	56

Figura 14. Pantalla Ver egresos: Los Autores.

En esta pantalla ingresar los datos requeridos y dar un clic en el botón **Añadir Productos**, se irán añadiendo cuantos productos se quieran ingresar para terminar dar clic en el botón **Finalizar**.

## 10. Reportes

Para obtener un reporte se debe dar clic en la opción de reporte que se desea, el detalle de los mismos se encuentra en opción Reportes Estadísticos, eligiendo cualquier opción se mostrará una pantalla desplegando la consulta correspondiente a la opción elegida, en la pantalla desplegada en la parte superior se visualiza tres opciones: 1) exportación de datos a pdf, 2) documento de Word y 3) documento de Excel. Se debe elegir el formato de documento que se desea, dar clic y ubicar el lugar en donde se guardará el archivo convertido.

Para realizar el reporte de la Planificación Operativa Anual hay que elegir el submenú Plan Anual, opción Reporte Plan Anual.





#### MOVIMIENTO POR DESTINATARIO

CODIGO	PROVEEDOR	ORGANIZACION	FECHA TRANS.
(2010)- 1	TANASA	Consejo Nacional de las Mujeres	2010-10-11
(2010)- 2	TANASA	Consejo Nacional de las Mujeres	2010-10-12
(2010)- 3	TANASA	Consejo Nacional de las Mujeres	2010-10-12
(2010)- 4	TANASA	Consejo Nacional de las Mujeres	2010-10-12
(2010)- 5	TANASA	Consejo Nacional de las Mujeres	2010-10-12
(2010)- 6	TANASA	Consejo Nacional de las Mujeres	2010-10-14
(2010)- 7	TANASA	Consejo Nacional de las Mujeres	2010-10-16
(2010)- 8	TANASA	Consejo Nacional de las Mujeres	2010-11-08
(2011)- 10	EDITORIAL LATINA	Consejo Nacional de las Mujeres	2011-03-17
(2011)- 11	EDITORIAL LATINA	Consejo Nacional de las Mujeres	2011-03-17



Figura 15. Pantalla Reportes: Los Autores.

## 11. Manual de Usuario

Al dar clic en esta opción se desplegará en una nueva pantalla un archivo pdf que contiene un manual de usuario para una mejor manipulación del sistema de control e inventario.

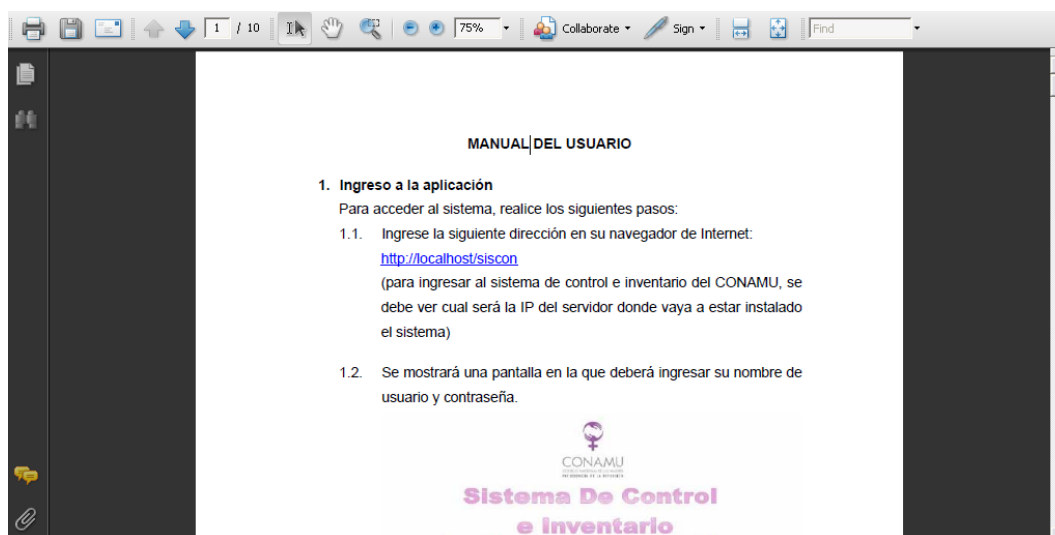


Figura 16. Pantalla Manual de Usuario: Los Autores

## CONTENIDO

	pág.
LISTA DE ILUSTRACIONES	III
LISTA DE TABLAS	V
<b>MANUAL TÉCNICO</b>	<b>1</b>
<b>Introducción</b>	<b>1</b>
<b>1. Descripción general</b>	<b>1</b>
1.1. Alcance	1
<b>2. Ingeniería del proyecto</b>	<b>2</b>
2.1. Modelo de requisitos	2
2.1.1. Modelo de casos de uso	3
2.1.2. Modelo de interfaces	3
2.1.3. Modelo de dominio	4
2.2. Modelo de análisis	5
2.2.1. Arquitectura de clases	5
2.2.2. Diccionario de clases	14
2.2.3. Diccionario de datos	26
2.2.4. Diagramas de secuencia	45
2.3. Modelo de diseño	45
2.3.1. Diseño de objetos	45
2.3.2. Diseño de interfaces	46
2.3.3. Diseño del sistema	47
2.3.4. Diseño de menús y mapas de navegación	48
2.3.5. Diseño de arquitectura del sistema	52
<b>3. Diagrama global del sistema</b>	<b>52</b>
3.1. Diagrama de componentes	52
3.2. Diagrama de distribución	53
<b>4. Instrucciones de instalación y ejecución</b>	<b>53</b>

	II
4.1. Requerimientos de hardware y software	53
<b>5. Seguridad y control</b>	<b>54</b>
5.1. Seguridad en el acceso de la información	54
5.2. Creación de usuarios	54
5.3. Seguridad de Acceso	55
5.4. Seguridad de usuario	55
5.5. Copias de seguridad	56
5.6. Lista de mensajes de error	56
<b>6. Código fuente</b>	<b>57</b>

## LISTA DE ILUSTRACIONES

	pág.
<b>Figura 1.</b> Modelo de dominio	4
<b>Figura 2.</b> Ingreso	5
<b>Figura 3.</b> Egresos	6
<b>Figura 4.</b> Evaluación	7
<b>Figura 5.</b> Conexión	7
<b>Figura 6.</b> Organización	8
<b>Figura 7.</b> Tipo de documento	8
<b>Figura 8.</b> Tipo de Unidad	9
<b>Figura 9.</b> Destino Geográfico	9
<b>Figura 10.</b> Funcionaria	10
<b>Figura 11.</b> Productos	10
<b>Figura 12.</b> Tipo de Producto	11
<b>Figura 13.</b> Objeto transacción	11
<b>Figura 14.</b> Proveedores	12
<b>Figura 15.</b> Tipo de Servicio	12
<b>Figura 16.</b> Usuario Final.	13
<b>Figura 17.</b> Catálogos de tablas básicas	45
<b>Figura 18.</b> Reportes Estadísticos	46
<b>Figura 19.</b> Transacciones	46
<b>Figura 20.</b> Plan Anual	46
<b>Figura 21.</b> Ayuda	46
<b>Figura 22.</b> Diseño Lógico	47
<b>Figura 23.</b> Diseño Físico	48
<b>Figura 24.</b> Diseño de Menú	49
<b>Figura 25.</b> Mapa de Navegación	49
<b>Figura 26.</b> Mapa de Navegación	50
<b>Figura 27.</b> Mapa de Navegación	50

	IV
<b>Figura 28.</b> Mapa de Navegación	51
<b>Figura 29.</b> Mapa de Navegación	51
<b>Figura 30.</b> Arquitectura del Sistema	52
<b>Figura 31.</b> Diagrama de componente	52
<b>Figura 32.</b> Diagrama de Distribución	53

**LISTA DE TABLAS**

	<b>pág.</b>
<b>Tabla 1.</b> Destinogeografico	14
<b>Tabla 2.</b> Organizacion	15
<b>Tabla 3.</b> Tipodocumento	16
<b>Tabla 4.</b> Tipounidad	17
<b>Tabla 5.</b> Objetotransaccion	18
<b>Tabla 6.</b> Tipoproducto	19
<b>Tabla 7.</b> Tiposervicio	20
<b>Tabla 8.</b> Usuariofinal	21
<b>Tabla 9.</b> Evaluacion	22
<b>Tabla 10.</b> Evaluacion	23
<b>Tabla 11.</b> Proveedores	24
<b>Tabla 12.</b> Validar_Usuario	25
<b>Tabla 13.</b> Funcionaria	26
<b>Tabla 14.</b> Detalle_ingreso	27
<b>Tabla 15.</b> Proveedor	28
<b>Tabla 16.</b> Egreso	29
<b>Tabla 17.</b> Producto	30
<b>Tabla 18.</b> Ingreso	31
<b>Tabla 19.</b> Objetivo	32
<b>Tabla 20.</b> Tipodocumento	33
<b>Tabla 21.</b> Tipoproducto	34
<b>Tabla 22.</b> Destinogeo	35
<b>Tabla 23.</b> Objetotransaccion	36
<b>Tabla 24.</b> Organizacion	37
<b>Tabla 25.</b> Privilegio	38
<b>Tabla 26.</b> Responsable	39
<b>Tabla 27.</b> Tipo_servicio	40

	VI
<b>Tabla 28.</b> Tipo_usuario	41
<b>Tabla 29.</b> TipoUnidad	42
<b>Tabla 30.</b> Detalle_Egreso	43
<b>Tabla 31.</b> Registro_user	44

# **MANUAL TÉCNICO**

## **INTRODUCCIÓN**

Este documento pretende ser una guía en la que se dará indicaciones de qué pasos realizar para modificar ciertos aspectos del sistema. Así como también de ciertas funcionalidades complejas del sistema que permiten mostrar una información más específica y seleccionada.

Estas indicaciones tienen como objetivo que el administrador que vaya a modificar y controlar los procesos o funciones del sistema, tenga una guía que le oriente y le ayude a llevar a cabo su tarea. Este manual no pretende ser algo definitivo, sino más bien orientativo, por lo que puede haber ciertos aspectos que sean recomendables modificar, o que no sean mencionados. No se abordan aquí todas las partes del sistema que pueden modificarse, sino aquellas más importantes.

## **1. DESCRIPCIÓN GENERAL**

En vista de las necesidades que se han suscitado en el Centro de Difusión y Documentación del Consejo Nacional de las Mujeres, para manejar los datos y obtener la información deseada se ha desarrollado el sistema con la finalidad de optimizar los datos arrojados de dicha Institución.

El sistema ayudará a procesar la información de una manera más sencilla y ágil, minimizando así el tiempo en las diferentes tareas a realizar.

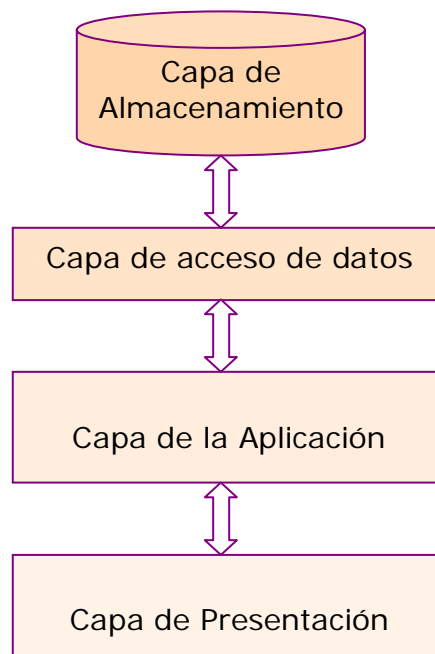
### **1.1. ALCANCE**

Los resultados obtenidos del desarrollo del sistema ayudarán a cubrir todas las falencias que se han presentado en el Centro de Difusión y Documentación del Consejo Nacional de las Mujeres, de esta manera se ha logrado un óptimo manejo de la información así como también presentar ciertas funciones y operaciones en el sistema que van a permitir al usuario obtener un mejor resultado de la información requerida de una forma mas específica y fácil.

Todos los módulos desarrollados que se presentan en el sistema cumplen con todos los requerimientos especificados por el usuario.



Las cuatro capas de las cuales estará compuesto el sistema se dividirán en lo siguiente<sup>1</sup>:



*Figura 1. Arquitectura 4 capas: Los autores.*

### **Capa de Almacenamiento**

Esta compuesta por la base de datos a utilizar para el proyecto como es de PostgreSQL.

### **Capa de acceso de datos**

Encargada de la manipulación de los datos que se gestionan en la aplicación.

### **Capa de Aplicación**

Aquí todos los procedimientos y procesos para transacciones de productos y generación de reportes.

### **Capa de Presentación**

Esta es la única capa visible para el usuario y la que interactúa con el cliente de tal manera que ofrece un ambiente agradable para trabajar además que provee la seguridad necesaria, ya que se encuentra separada de las demás capas.

## **2. INGENIERÍA DEL PROYECTO**

### **2.1. MODELO DE REQUISITOS**

---

<sup>1</sup> Documento de Arquitectura de Diagramas.

[En línea]. <http://www.scribd.com/Documento-de-Arquitectura-Diagramas/d/10074807>

### 2.1.1. MODELO DE CASOS DE USO

Para poder visualizar los modelos de casos de uso dirigirse al Capítulo 4 páginas 70- 85 .

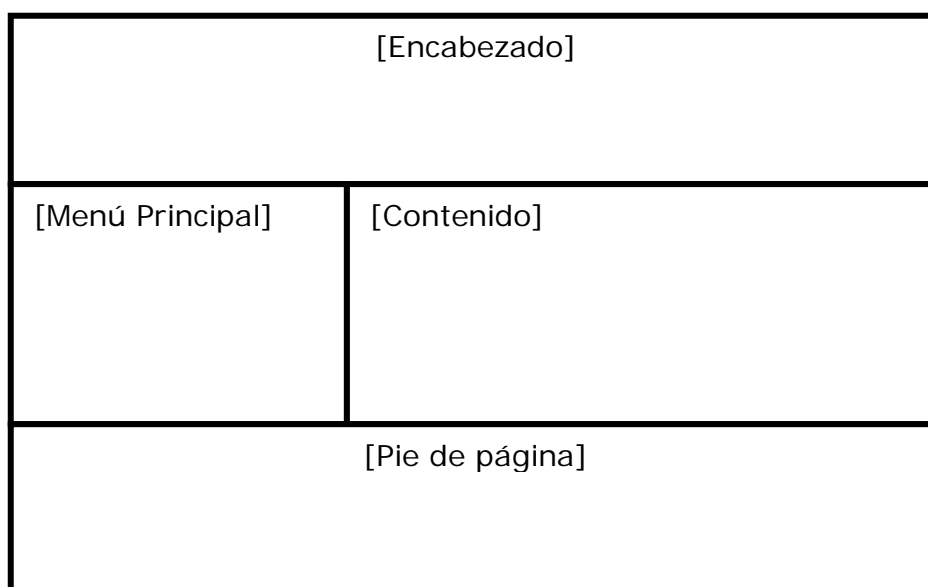
### 2.1.2. MODELO DE INTERFACES

Las interfaces del sistema que a continuación se van a diseñar es acorde a las especificaciones y estándares emitido por las funcionarias.

La interfaz que se maneja en el sistema es única, ya que consta de el Encabezado donde se ubica el logo de la Institución y el titulo del sistema. Dentro del Menú Principal se encuentra el menú desplegable con todas las opciones que maneja el sistema.

En la parte de Contenido se muestra el formulario en el momento que se elija alguna opción del menú.

Pie de página se ubicará el nombre de la Institución, dirección y sus respectivos teléfonos.



*Figura 2. Modelo de interfaces: Los Autores.*

### 2.1.3. MODELO DE DOMINIO

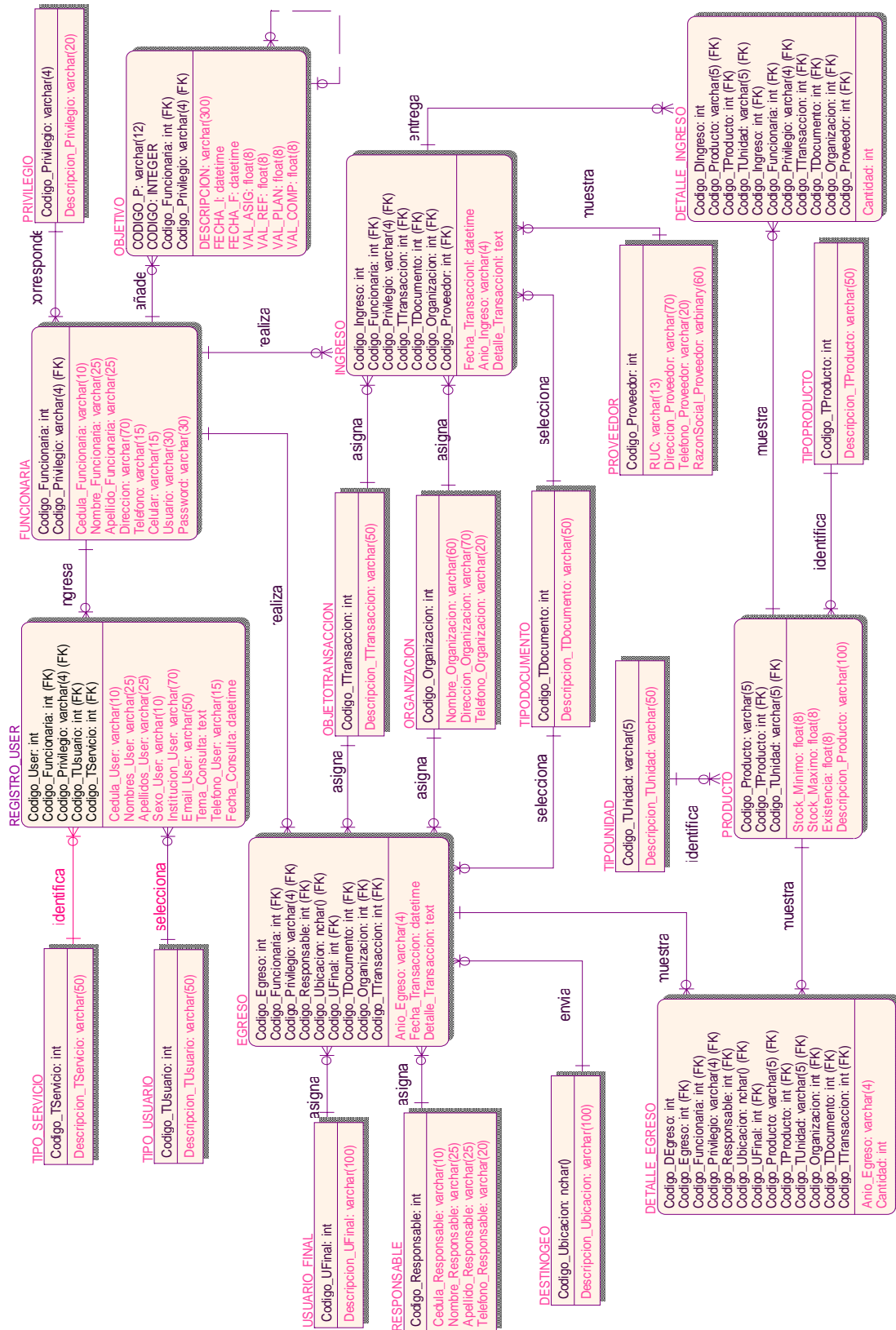


Figura 3. Modelo de dominio: Los Autores.

## 2.2. MODELO DE ANÁLISIS

### 2.2.1. ARQUITECTURA DE CLASES

#### Ingresos

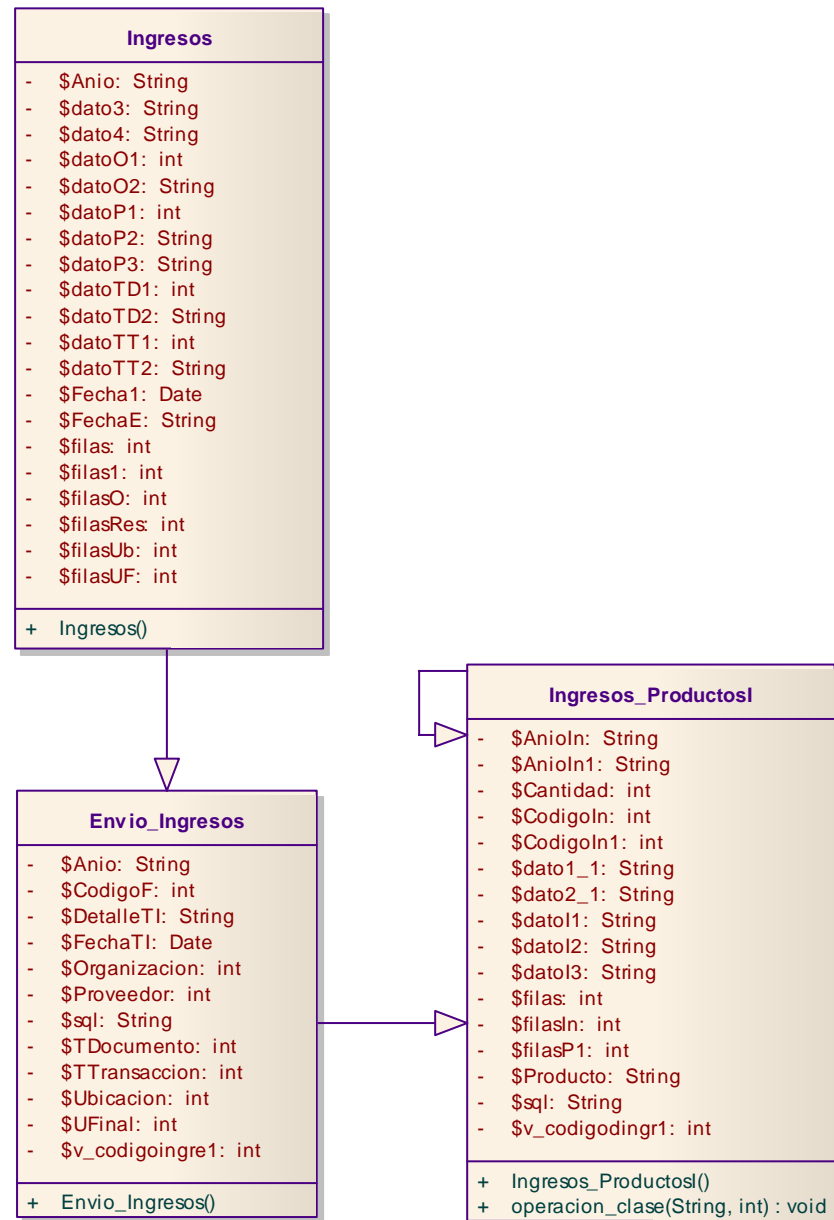


Figura 4. Diagrama: Ingreso. Los Autores.

## Egresos

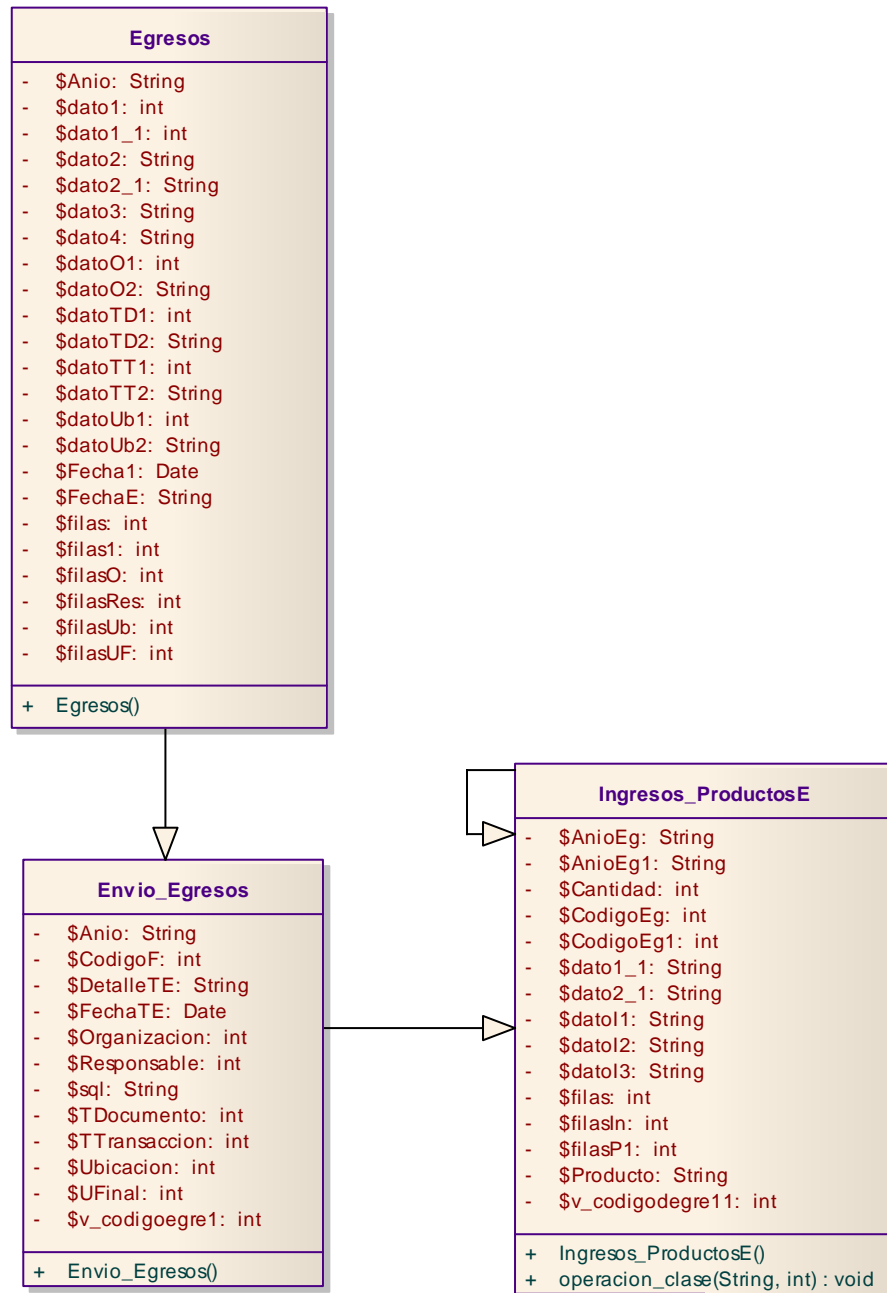


Figura 5. Diagrama: Egresos. Los Autores.

## Evaluación

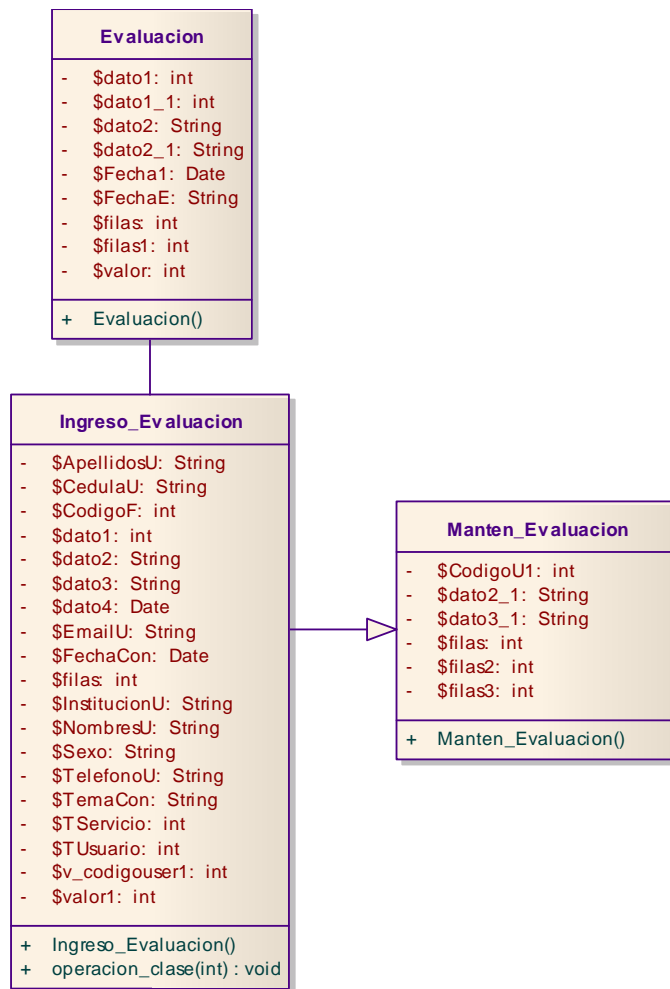


Figura 6. Diagrama: Evaluación. Los Autores.

## Conexión

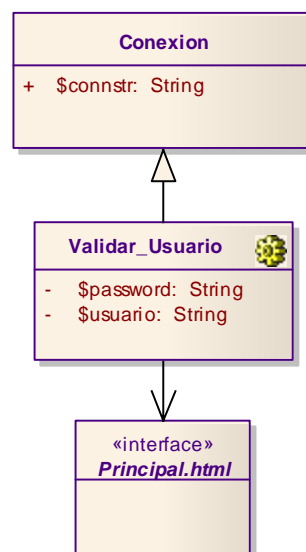


Figura 7. Diagrama: Conexión. Los Autores.

## Organización

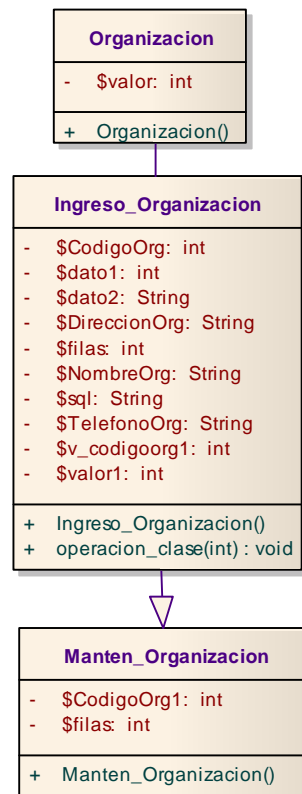


Figura 8. Diagrama: Organización. Los Autores.

## Tipo de documento

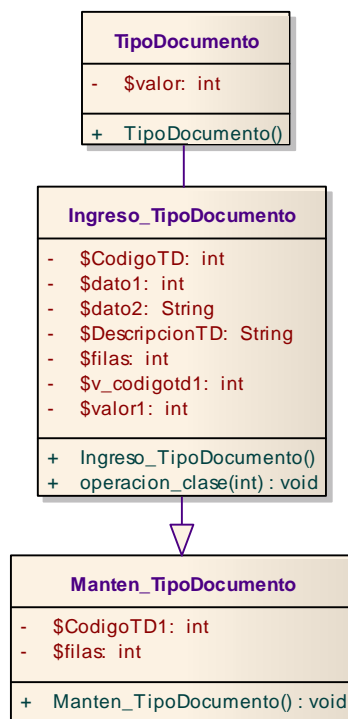


Figura 9. Diagrama: Tipo de documento. Los Autores.

## Tipo de Unidad



Figura 10. Diagrama: Tipo de Unidad. Los Autores.

## Destino Geográfico

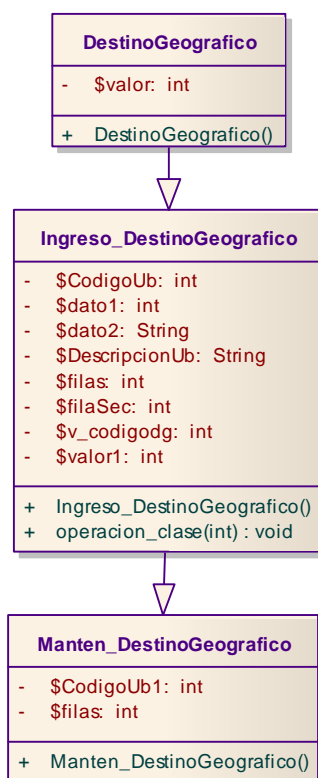


Figura 11. Diagrama: Destino Geográfico. Los Autores.



## Funcionaria

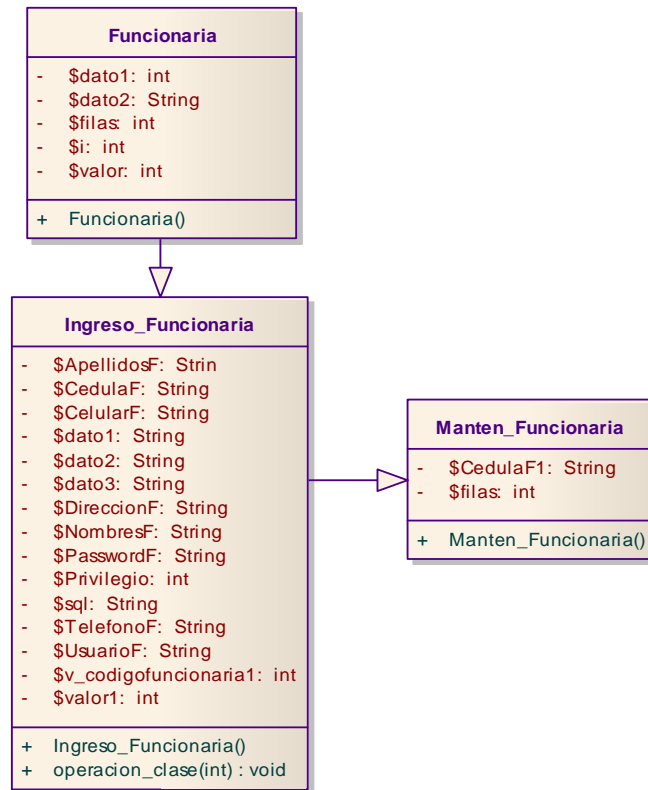


Figura 12. Diagrama: Funcionaria. Los Autores.

## Productos

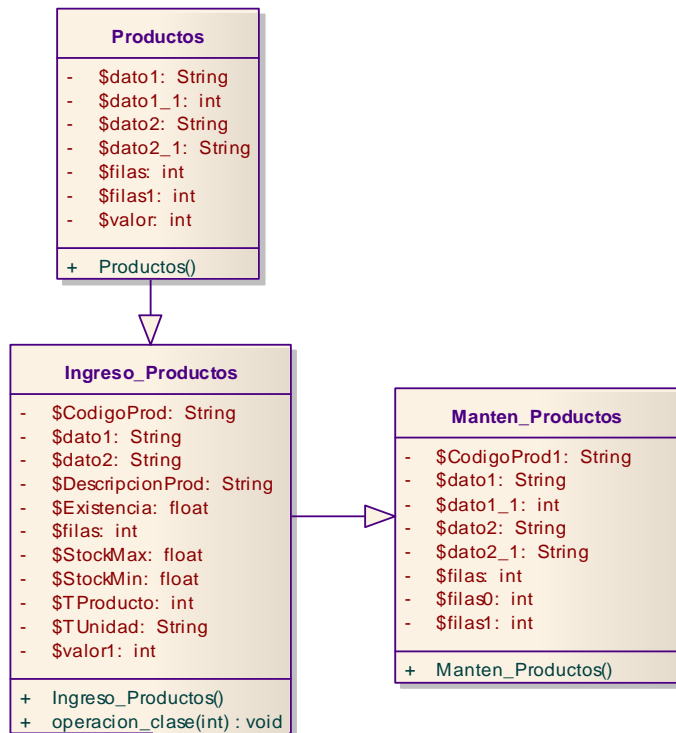


Figura 13. Diagrama: Productos. Los Autores.

## Tipo de Producto

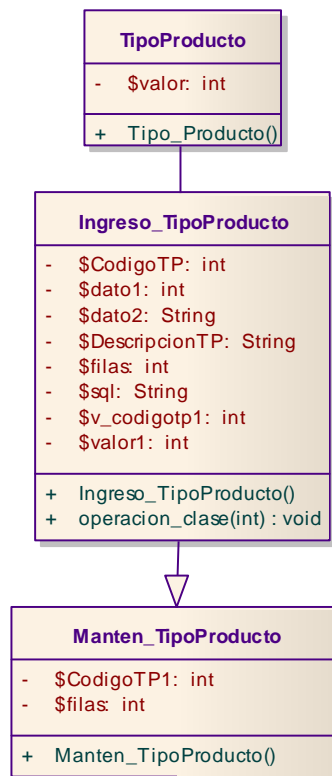


Figura 14. Diagrama: Tipo de Producto. Los Autores.

## Objeto transacción

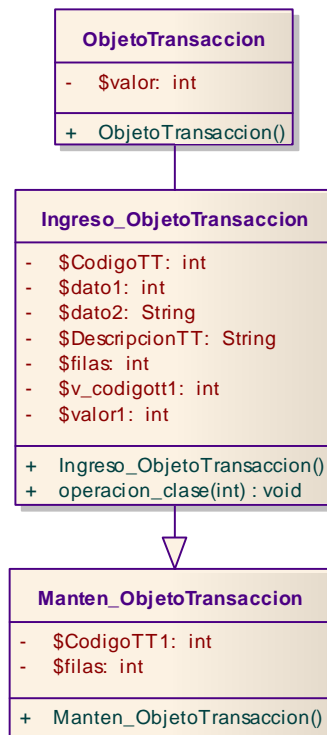


Figura 15. Diagrama: Objeto transacción. Los Autores.

## Proveedores

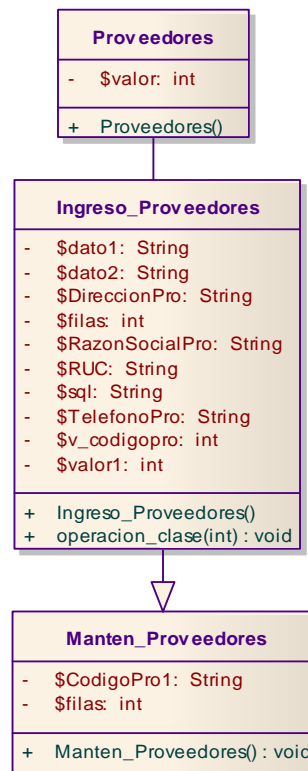


Figura 16. Diagrama: Proveedores. Los Autores.

## Tipo de servicio

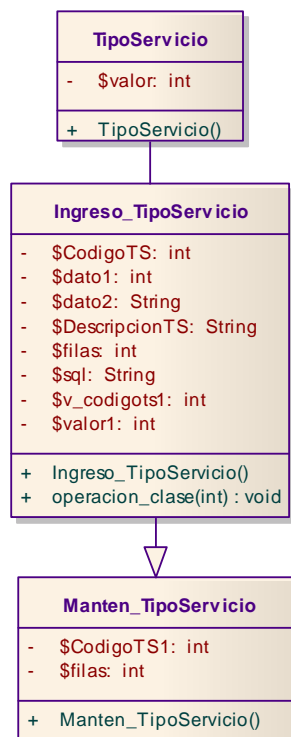


Figura 17. Diagrama: Tipo de Servicio. Los Autores.

## Usuario Final

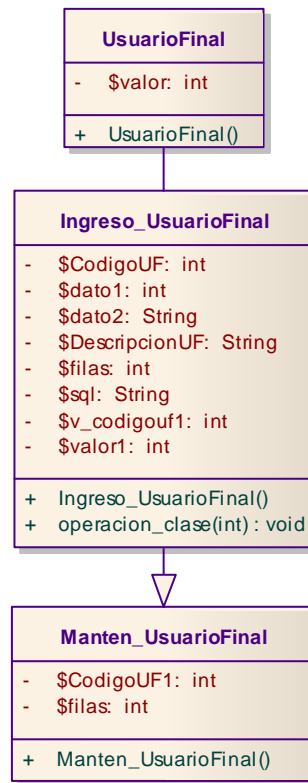


Figura 18. Diagrama: Usuario Final. Los Autores.

## 2.2.2. DICCIONARIO DE CLASES

CLASE: DESTINO GEOGRAFICO		
Nombre	Tipo	Descripción
\$valor	int	Valor que se envía a subclase Ingreso_DestinoGeografico para realizar la operación asignada según valor, sea Ingreso (1), Modificación (2) o Búsqueda (3)
SUBCLASE: INGRESO_DESTINO GEOGRAFICO		
\$CodigoUb	int	Variable donde se asigna el dato para el código de ubicación.
\$dato1	Int	Variable donde se asigna el dato para el código de ubicación para ser mostrado en tabla de resultados
\$dato2	String	Variable donde se asigna el dato para la descripción de ubicación para ser mostrado en tabla de resultados
\$DescripcionUb	Int	Variable donde se asigna el dato para la descripción de ubicación.
\$filas	Int	Variable donde se almacena el número total de registros hallados según consulta establecida.
\$filasSec	Int	Variable donde se almacena el número total de registros hallados según consulta para establecer número secuencial.
\$v_codigodg	Int	Variable para asignar un valor secuencial que irá para el código de ubicación
\$valor1	Int	Variable donde se asigna el dato proveniente de la clase DestinoGeografico.
operacion_clase()	int	Este método recibe un dato entero (1, 2 o 3) y según su valor realizará las operaciones de ingreso, actualización o búsqueda, no devuelve ningún valor.
SUBCLASE: MANTEN_DESTINO GEOGRAFICO		
\$CodigoUb1	Int	Variable donde se asigna el dato para el código de ubicación. Para proceder a realizar consulta y mostrar resultado para su posterior modificación
\$filas	int	Variable donde se almacena el número total de registros hallados según consulta establecida.

Tabla1. Diccionario de Clase: (Destinogeografico): Los Autores.

CLASE: ORGANIZACION		
Nombre	Tipo	Descripción
\$valor	int	Valor que se envía a subclase Ingreso_Organización para realizar la operación asignada según valor, sea Ingreso (1), Modificación (2) o Búsqueda (3)
<b>SUBCLASE: INGRESO_ORGANIZACION</b>		
\$CodigoOrg	int	Variable donde se asigna el dato para el código de la organización.
\$dato1	Int	Variable donde se asigna el dato para el código de organización para ser mostrado en tabla de resultados
\$dato2	String	Variable donde se asigna el dato para la descripción de organización para ser mostrado en tabla de resultados
\$DireccionOrg	String	Variable donde se asigna el dato para la dirección de la organización.
\$filas	Int	Variable donde se almacena el número total de registros hallados según consulta establecida.
\$NombreOrg	String	Variable donde se almacena el número total de registros hallados según consulta para establecer número secuencial.
\$sql	String	Variable donde se almacena la sentencia SQL para ser ejecutada.
\$TelefonoOrg	String	Variable donde se asigna el dato para el teléfono de la organización.
\$v_codigoorg1	Int	Variable para asignar un valor secuencial que irá para el código de organización.
\$valor1	Int	Variable donde se asigna el dato proveniente de la clase Organización.
operacion_clase()	int	Este método recibe un dato entero (1, 2 o 3) y según su valor realizará las operaciones de ingreso, actualización o búsqueda, no devuelve ningún valor.
<b>SUBCLASE: MANTEN_ORGANIZACION</b>		
\$CodigoOrg1	Int	Variable donde se asigna el dato para el código de la organización. Para proceder a realizar consulta y mostrar resultado para su posterior modificación
\$filas	int	Variable donde se almacena el número total de registros hallados según consulta establecida.

Tabla2. Diccionario de Clase: (Organización): Los Autores.

CLASE: TIPODOCUMENTO		
Nombre	Tipo	Descripción
\$valor	int	Valor que se envía a subclase Ingreso_TipoDocumento para realizar la operación asignada según valor, sea Ingreso (1), Modificación (2) o Búsqueda (3)
<b>SUBCLASE: INGRESO_ TIPODOCUMENTO</b>		
\$CodigoTD	int	Variable donde se asigna el dato para el código del tipo de documento.
\$dato1	Int	Variable donde se asigna el dato para el código del tipo de documento para ser mostrado en tabla de resultados
\$dato2	String	Variable donde se asigna el dato para la descripción del tipo de documento para ser mostrado en tabla de resultados
\$DescripcionTD	String	Variable donde se asigna el dato para la descripción del tipo de documento.
\$filas	Int	Variable donde se almacena el número total de registros hallados según consulta establecida.
\$v_codigoootd1	Int	Variable para asignar un valor secuencial que irá para el código del tipo de documento.
\$valor1	Int	Variable donde se asigna el dato proveniente de la clase TipoDocumento.
operacion_clase()	int	Este método recibe un dato entero (1, 2 o 3) y según su valor realizará las operaciones de ingreso, actualización o búsqueda, no devuelve ningún valor.
<b>SUBCLASE: MANTEN_ TIPODOCUMENTO</b>		
\$CodigoTD1	Int	Variable donde se asigna el dato para el código del tipo de documento. Para proceder a realizar consulta y mostrar resultado para su posterior modificación
\$filas	int	Variable donde se almacena el número total de registros hallados según consulta establecida.

Tabla3. Diccionario de Clase: (Tipodocumento): Los Autores.

CLASE: TIPOUNIDAD		
Nombre	Tipo	Descripción
\$valor	int	Valor que se envía a subclase Ingreso_TipoUnidad para realizar la operación asignada según valor, sea Ingreso (1), Modificación (2) o Búsqueda (3)
SUBCLASE: INGRESO_TIPOUNIDAD		
\$CodigoTU	String	Variable donde se asigna el dato para el código del tipo de unidad.
\$dato1	String	Variable donde se asigna el dato para el código del tipo de unidad para ser mostrado en tabla de resultados
\$dato2	String	Variable donde se asigna el dato para la descripción del tipo de unidad para ser mostrado en tabla de resultados
\$DescripcionTU	String	Variable donde se asigna el dato para la descripción del tipo de unidad.
\$filas	Int	Variable donde se almacena el número total de registros hallados según consulta establecida.
\$sql	String	Variable donde se almacena la sentencia SQL para ser ejecutada.
\$valor1	Int	Variable donde se asigna el dato proveniente de la clase TipoUnidad.
operacion_clase()	int	Este método recibe un dato entero (1, 2 o 3) y según su valor realizará las operaciones de ingreso, actualización o búsqueda, no devuelve ningún valor.
SUBCLASE: MANTEN_TIPOUNIDAD		
\$CodigoTU1	String	Variable donde se asigna el dato para el código del tipo de unidad. Para proceder a realizar consulta y mostrar resultado para su posterior modificación
\$filas	int	Variable donde se almacena el número total de registros hallados según consulta establecida.

Tabla4. Diccionario de Clase: (Tipounidad): Los Autores.



CLASE: OBJETOTRANSACCION		
Nombre	Tipo	Descripción
\$valor	int	Valor que se envía a subclase Ingreso_ObjetoTransaccion para realizar la operación asignada según valor, sea Ingreso (1), Modificación (2) o Búsqueda (3)
<b>SUBCLASE: INGRESO_ OBJETO TRANSACCION</b>		
\$CodigoTT	int	Variable donde se asigna el dato para el código del objeto de transacción.
\$dato1	Int	Variable donde se asigna el dato para el código del objeto de transacción para ser mostrado en tabla de resultados
\$dato2	String	Variable donde se asigna el dato para la descripción del objeto de transacción para ser mostrado en tabla de resultados
\$DescripcionTT	String	Variable donde se asigna el dato para la descripción del tipo de transacción.
\$filas	Int	Variable donde se almacena el número total de registros hallados según consulta establecida.
\$v_codigooott1	Int	Variable para asignar un valor secuencial que irá para el código del objeto de transacción.
\$valor1	Int	Variable donde se asigna el dato proveniente de la clase ObjetoTransaccion.
operacion_clase()	int	Este método recibe un dato entero (1, 2 o 3) y según su valor realizará las operaciones de ingreso, actualización o búsqueda, no devuelve ningún valor.
<b>SUBCLASE: MANTEN_ OBJETO TRANSACCION</b>		
\$CodigoTT1	Int	Variable donde se asigna el dato para el código del objeto de transacción. Para proceder a realizar consulta y mostrar resultado para su posterior modificación
\$filas	int	Variable donde se almacena el número total de registros hallados según consulta establecida.

Tabla5. Diccionario de Clase: (Objetotransaccion): Los Autores.

CLASE: TIPOPRODUCTO		
Nombre	Tipo	Descripción
\$valor	int	Valor que se envía a subclase Ingreso_TipoProducto para realizar la operación asignada según valor, sea Ingreso (1), Modificación (2) o Búsqueda (3)
SUBCLASE: INGRESO_ TIPOPRODUCTO		
\$CodigoTP	int	Variable donde se asigna el dato para el código del tipo de producto.
\$dato1	Int	Variable donde se asigna el dato para el código del tipo de producto para ser mostrado en tabla de resultados
\$dato2	String	Variable donde se asigna el dato para la descripción del tipo de producto para ser mostrado en tabla de resultados
\$DescripcionTP	String	Variable donde se asigna el dato para la descripción del tipo de producto.
\$filas	Int	Variable donde se almacena el número total de registros hallados según consulta establecida.
\$sql	String	Variable donde se almacena la sentencia SQL para ser ejecutada.
\$v_codigoootp1	Int	Variable para asignar un valor secuencial que irá para el código del tipo de producto.
\$valor1	Int	Variable donde se asigna el dato proveniente de la clase TipoProducto.
operacion_clase()	int	Este método recibe un dato entero (1, 2 o 3) y según su valor realizará las operaciones de ingreso, actualización o búsqueda, no devuelve ningún valor.
SUBCLASE: MANTEN_ TIPOPRODUCTO		
\$CodigoTP1	Int	Variable donde se asigna el dato para el código del tipo de producto. Para proceder a realizar consulta y mostrar resultado para su posterior modificación
\$filas	int	Variable donde se almacena el número total de registros hallados según consulta establecida.

Tabla6. Diccionario de Clase: (Tipoproducto): Los Autores.

CLASE: TIPOSERVICIO		
Nombre	Tipo	Descripción
\$valor	int	Valor que se envía a subclase Ingreso_TipoServicio para realizar la operación asignada según valor, sea Ingreso (1), Modificación (2) o Búsqueda (3)
SUBCLASE: INGRESO_ TIPOSERVICIO		
\$CodigoTS	int	Variable donde se asigna el dato para el código del tipo de servicio.
\$dato1	Int	Variable donde se asigna el dato para el código del tipo de servicio para ser mostrado en tabla de resultados
\$dato2	String	Variable donde se asigna el dato para la descripción del tipo de servicio para ser mostrado en tabla de resultados
\$DescripcionTS	String	Variable donde se asigna el dato para la descripción del tipo de servicio.
\$filas	Int	Variable donde se almacena el número total de registros hallados según consulta establecida.
\$sql	String	Variable donde se almacena la sentencia SQL para ser ejecutada.
\$v_codigooots1	Int	Variable para asignar un valor secuencial que irá para el código del tipo de servicio.
\$valor1	Int	Variable donde se asigna el dato proveniente de la clase TipoServicio.
operacion_clase()	int	Este método recibe un dato entero (1, 2 o 3) y según su valor realizará las operaciones de ingreso, actualización o búsqueda, no devuelve ningún valor.
SUBCLASE: MANTEN_ TIPOSERVICIO		
\$CodigoTS1	Int	Variable donde se asigna el dato para el código del tipo de servicio. Para proceder a realizar consulta y mostrar resultado para su posterior modificación
\$filas	int	Variable donde se almacena el número total de registros hallados según consulta establecida.

Tabla7. Diccionario de Clase: (Tiposervicio): Los Autores.

CLASE: USUARIOFINAL		
Nombre	Tipo	Descripción
\$valor	int	Valor que se envía a subclase Ingreso_usuarioFinal para realizar la operación asignada según valor, sea Ingreso (1), Modificación (2) o Búsqueda (3)
SUBCLASE: INGRESO_usuarioFINAL		
\$CodigoUF	int	Variable donde se asigna el dato para el código del usuario final.
\$dato1	Int	Variable donde se asigna el dato para el código del usuario final para ser mostrado en tabla de resultados
\$dato2	String	Variable donde se asigna el dato para la descripción del usuario final para ser mostrado en tabla de resultados
\$DescripcionUF	String	Variable donde se asigna el dato para la descripción del usuario final.
\$filas	Int	Variable donde se almacena el número total de registros hallados según consulta establecida.
\$sql	String	Variable donde se almacena la sentencia SQL para ser ejecutada.
\$v_codigoouf1	Int	Variable para asignar un valor secuencial que irá para el código del usuario final.
\$valor1	Int	Variable donde se asigna el dato proveniente de la clase UsuarioFinal .
operacion_clase()	int	Este método recibe un dato entero (1, 2 o 3) y según su valor realizará las operaciones de ingreso, actualización o búsqueda, no devuelve ningún valor.
SUBCLASE: MANTEN_usuarioFINAL		
\$CodigoUF1	Int	Variable donde se asigna el dato para el código del usuario final. Para proceder a realizar consulta y mostrar resultado para su posterior modificación
\$filas	int	Variable donde se almacena el número total de registros hallados según consulta establecida.

Tabla8. Diccionario de Clase: (Usuariofinal): Los Autores.

CLASE: EVALUACION		
Nombre	Tipo	Descripción
\$dato1	Int	Variable donde se asigna el dato para el código del tipo de usuario para ser mostrado en tabla de resultados
\$dato1_1	Int	Variable donde se asigna el dato para la descripción del tipo de usuario para ser mostrado en tabla de resultados
\$dato2	String	Variable donde se asigna el dato para el código del servicio para ser mostrado en tabla de resultados
\$dato2_1	String	Variable donde se asigna el dato para el código del servicio para ser mostrado en tabla de resultados
\$Fecha1	Date	Variable para almacenar la fecha actual.
\$FechaE	String	Variable para ubicar la fecha conseguida en el formato "año-mes-día"
\$filas	Int	Variable donde se almacena el número total de registros hallados según consulta establecida.
\$filas1	Int	Variable donde se almacena el número total de registros hallados según consulta establecida.
\$valor	Int	Valor que se envía a subclase Ingreso_Evaluacion para realizar la operación asignada según valor, sea Ingreso (1), Modificación (2) o Búsqueda (3)
SUBCLASE: INGRESO_EVALUACION		
\$ApellidosU	String	Variable donde se asigna el dato para el Apellido del Usuario.
\$CedulaU	String	Variable donde se asigna el dato para la cedula del Usuario.
\$CodigoF	Int	Variable donde se asigna el dato para el código de la funcionaria que atendió.
\$dato1	Int	Variable donde se asigna el dato para el código del usuario para ser mostrado en tabla de resultados
\$dato2	String	Variable donde se asigna el dato para el nombre del usuario para ser mostrado en tabla de resultados
\$dato3	String	Variable donde se asigna el dato para el apellido del usuario para ser mostrado en tabla de resultados
\$dato4	Date	Variable donde se asigna el dato para la fecha de consulta.
\$EmailU	String	Variable donde se asigna el dato para el mail del usuario.

Tabla9. Diccionario de Clase: (Evaluacion): Los Autores.

\$FechaCon	Date	Variable donde se asigna el dato para la fecha de consulta.
\$filas	Int	Variable donde se almacena el número total de registros hallados según consulta establecida.
\$InstitucionU	String	Variable donde se asigna el dato para la institución del cual proviene el usuario.
\$NombresU	String	Variable donde se asigna el dato para el nombre del usuario.
\$Sexo	String	Variable donde se asigna el dato para el sexo (masculino o femenino) del usuario.
\$TelefonoU	String	Variable donde se asigna el dato para el teléfono del usuario.
\$TemaCon	String	Variable donde se asigna el dato para el tema de consulta.
\$TServicio	Int	Variable donde se asigna el dato para la descripción del tipo de servicio que se ofreció a la usuaria/o (mail, teléfono, verbal, etc.).
\$TUsuario	int	Variable donde se asigna el dato para el tipo de usuaria/o (colegio, universidad, empleada/o publica/o, etc.).
\$v_codigouser1	Int	Variable para asignar un valor secuencial que irá para el código del usuario.
\$valor1	Int	Variable donde se asigna el dato proveniente de la clase Evaluación.
operacion_clase()	int	Este método recibe un dato entero (1, 2 o 3) y según su valor realizará las operaciones de ingreso, actualización o búsqueda, no devuelve ningún valor.
<b>SUBCLASE: MANTEN_EVALUACION</b>		
\$CodigoU1	Int	Variable donde se asigna el dato para el código del usuario. Para proceder a realizar consulta y mostrar resultado para su posterior modificación
\$dato2_1	String	Variable donde se asigna el dato para la descripción del tipo de usuario para ser mostrado en tabla de resultados
\$dato3_1	String	Variable donde se asigna el dato para la descripción del tipo de servicio para ser mostrado en tabla de resultados
\$filas	Int	Variable donde se almacena el número total de registros hallados según consulta establecida.
\$filas2	Int	Variable donde se almacena el número total de registros hallados según consulta establecida.
\$filas3	Int	Variable donde se almacena el número total de registros hallados según consulta establecida.

Tabla9. Diccionario de Clase: (Evaluacion): Los Autores.

CLASE: PROVEEDORES		
Nombre	Tipo	Descripción
\$valor	int	Valor que se envía a subclase Ingreso_Proveedores para realizar la operación asignada según valor, sea Ingreso (1), Modificación (2) o Búsqueda (3)
<b>SUBCLASE: INGRESO_PROVEEDORES</b>		
\$dato1	String	Variable donde se asigna el dato para el RUC del proveedor para ser mostrado en tabla de resultados
\$dato2	String	Variable donde se asigna el dato para la la razón social del proveedor para ser mostrado en tabla de resultados
\$DireccionPro	String	Variable donde se asigna el dato para la dirección del proveedor.
\$filas	Int	Variable donde se almacena el número total de registros hallados según consulta establecida.
\$sql	String	Variable donde se almacena la sentencia SQL para ser ejecutada.
\$RazonSocialPro	String	Variable donde se asigna el dato para el nombre o Razón Social del proveedor.
\$RUC	String	Variable donde se asigna el dato para el RUC del proveedor.
\$TelefonoPro	String	Variable donde se asigna el dato para el teléfono del proveedor.
\$v_codigoopro	Int	Variable para asignar un valor secuencial que irá para el código del proveedor.
\$valor1	Int	Variable donde se asigna el dato proveniente de la clase UsuarioFinal .
operacion_clase()	int	Este método recibe un dato entero (1, 2 o 3) y según su valor realizará las operaciones de ingreso, actualización o búsqueda, no devuelve ningún valor.
<b>SUBCLASE: MANTEN_PROVEEDORES</b>		
\$CodigoPro1	Int	Variable donde se asigna el dato para el código del proveedor. Para proceder a realizar consulta y mostrar resultado para su posterior modificación
\$filas	int	Variable donde se almacena el número total de registros hallados según consulta establecida.

Tabla10. Diccionario de Clase: (Proveedores): Los Autores.

CLASE: VALIDAR_USUARIO		
Nombre	Tipo	Descripción
\$password	String	Variable donde se asigna el dato del password de la funcionaria.
\$usuario	String	Variable donde se asigna el dato del nombre de usuario de la funcionaria.
SUBCLASE: CONEXIÓN		
\$connstr	String	Variable donde se almacena la sentencia SQL para ser ejecutada.

Tabla11. Diccionario de Clase: (Validar\_Usuario): Los Autores.



## 2.2.3. DICCIONARIO DE DATOS

<b>TABLA: Funcionaria</b> En esta tabla se almacenarán la información correspondiente a la funcionaria como son: Cédula, nombres completos, dirección, teléfono convencional y celular, su respectivo nombre de usuario y password además de su respectivo perfil de usuario.				
Nombre	Tipo	Tamaño	Descripción	Tipo de Restricción
Codigo_Funcionaria	int	4	Contiene el código que identifica a la funcionaria.	pk, No nulo.
Cedula_Funcionaria	varchar	10	Contiene el numero de cedula funcionaria	ck.
Nombre_Funcionaria	varchar	25	Contiene los nombres de la funcionaria.	ck.
Apellido_Funcionaria	varchar	25	Contiene los apellidos de la funcionaria.	ck.
Direccion	varchar	70	Contiene la dirección de la funcionaria.	No nulo.
Telefono	varchar	15	Contiene el teléfono de la funcionaria.	No nulo.
Celular	varchar	15	Contiene el celular de la funcionaria.	No nulo.
Usuario	varchar	30	Contiene el Nick de usuario.	ck.
Password	varchar	30	Contiene la contraseña del usuario.	ck.
Codigo_Privilegio	varchar	4	Contiene el código de privilegio para ingresar al sistema.	fk
<b>Clave primaria</b>	Codigo_Funcionaria			
<b>Clave foránea</b>	Codigo_Privilegio			
<b>Check</b>	Sean números			
	Este lleno el campo			
	Este lleno el campo			
	Este lleno el campo			
	Este lleno el campo			

Tabla12. Estructura para la tabla Funcionaria (Funcionaria): Los Autores.

<b>TABLA: Detalle_Ingreso</b> En esta tabla se almacenarán la información correspondiente al detalle de los productos que se reciben o son ingresados al sistema sean por devolución, compra, donación, etc.				
Nombre	Tipo	Tamaño	Descripción	Tipo de Restricción
Codigo_DIngreso	Int	8	Contiene el código que identifica al detalle de ingreso.	pk. No nulo.
Codigo_Ingreso	int	8	Contiene el código que identifica al los ingresos.	fk.
Anio_Ingreso	varchar	4	Contiene el código que identifica el año del ingreso.	fk.
Codigo_Producto	varchar	5	Contiene el código que identifica a los productos.	fk.
Cantidad	int	8	Contiene el código que identifica cantidad del ingreso.	ck.
<b>Clave primaria</b>		Codigo_DIngreso		
<b>Clave foránea</b>		Codigo_Ingreso	<b>Tabla de referencia</b>	Ingreso
		Anio_Ingreso	<b>Tabla de referencia</b>	Ingreso
		Codigo_Producto	<b>Tabla de referencia</b>	Producto
<b>Check</b>		Cantidad	Cantidad > 0	

Tabla13. Estructura para la tabla Detalle de ingreso (Detalle\_ingreso): Los Autores.

<b>TABLA: Proveedor</b> En esta tabla se almacenarán la información correspondiente a los datos personales o institucionales de cada proveedor que tenga la institución.				
Nombre	Tipo	Tamaño	Descripción	Tipo de Restricción
Codigo_Proveedor	int	4	Código que identifica al proveedor.	pk.
RUC	varchar	13	Contiene el RUC del proveedor	ck.
RazonSocial_Proveedor	varchar	60	Contiene el nombre del proveedor.	ck.
Direccion_Proveedor	varchar	70	Contiene la dirección del proveedor.	No nulo.
Telefono_Proveedor	varchar	20	Contiene el teléfono del proveedor.	No nulo.
<b>Clave primaria</b>	Codigo_Proveedor			
<b>Clave foránea</b>			<b>Tabla de referencia</b>	
<b>Check</b>	RUC		Dato numérico > a 13 dígitos	
	RazonSocial_Proveedor		Campo no debe estar vacío	

Tabla14. Estructura para la tabla Proveedor (Proveedor): Los Autores.

<b>TABLA: Egreso</b> En esta tabla se almacenarán la información correspondiente a los datos de cabecera para los egresos que se realizan.				
Nombre	Tipo	Tamaño	Descripción	Tipo de Restricción
Codigo_Egreso	int	8	Código generado secuencialmente usando las dos primeras letras de la transacción seguido de un número secuencial.	pk No nulo.
Anio_Egreso	varchar	4	Contiene el año del egreso.	ck.
Codigo_Responsable	int	4	Contiene el código que identifica al responsable.	fk.
Codigo_UFinal	int	4	Contiene el código que identifica al usuario final.	fk.
Codigo_Ubicacion	int	4	Contiene el código que identifica a la ubicación.	fk.
Codigo_Organizacion	int	4	Contiene el código que identifica a la organización.	fk.
Fecha_Transaccion	date		Contiene la fecha del egreso del evento.	ck.
Detalle_Transaccion	text		Contiene el detalle descriptivo de la transacción.	nulo.
Codigo_Funcionaria	int	4	Contiene el código que identifica a la funcionaria.	fk.
Codigo_TDocumento	int	4	Contiene el código que identifica el tipo de documento.	fk.
Codigo_TTransaccion	int	4	Contiene el código que identifica el tipo de transacción.	fk.
<b>Clave primaria</b>	Codigo_Egreso			
<b>Clave foránea</b>	Codigo_Responsable		<b>Tabla de referencia</b>	Responsable
	Codigo_UFinal		<b>Tabla de referencia</b>	Usuario_Final
	Codigo_Ubicacion		<b>Tabla de referencia</b>	DestinoGeo
	Codigo_Organizacion		<b>Tabla de referencia</b>	Organización
	Codigo_Funcionaria		<b>Tabla de referencia</b>	Funcionaria
	Codigo_TDocumento		<b>Tabla de referencia</b>	Tipo_Documento
	Codigo_TTransaccion		<b>Tabla de referencia</b>	Tipo_Transaccion
<b>Check</b>	Anio_Egreso		Número 4 dígitos	
	Fecha_Transaccion		Tipo Date Formato dd/mm/aa	

Tabla15. Estructura para la tabla Egreso (Egreso): Los Autores.

TABLA: Producto				
En esta tabla se registrarán la información correspondiente de cada producto existente para ser difundido o donado.				
Nombre	Tipo	Tamaño	Descripción	Tipo de Restricción
Codigo_Producto	varchar	5	Código de producto generado las dos primeras letras del tipo del producto seguido de un numero secuencial	pk
Stock_Minimo	float	8	Contiene las unidades existentes mínimas.	ck.
Stock_Maximo	float	8	Contiene las unidades existentes máximas.	ck.
Existencia	float	8	Contiene la cantidad existente de artículos o productos en bodega.	ck.
Codigo_TUnidad	varchar	5	Contiene el código de tipo de unidad.	fk.
Codigo_TPProducto	int	4	Contiene el código de tipo de producto.	fk.
Descripcion_producto	varchar	100	Indica si el producto es un folleto, libro etc.	No nulo.
<b>Clave primaria</b>	Codigo_Producto			
<b>Clave foránea</b>	Codigo_TUnidad		<b>Tabla de referencia</b>	Tipo_Unidad
	Codigo_TPProducto		<b>Tabla de referencia</b>	Tipo_Producto
<b>Check</b>	Stock_Minimo		Stock_Minimo > 5	
	Stock_Maximo		Stock_Maximo menor a cierta cantidad dependiendo el producto	
	Existencia		Ir aumentando o disminuyendo dependiendo los egresos o ingresos	

Tabla16. Estructura para la tabla Producto (Producto): Los Autores.

<b>TABLA: Ingreso</b> En esta tabla se almacenarán la información correspondiente al detalle de los productos que se recepan o son ingresados al sistema sean por devolución, compra, donación, etc.				
Nombre	Tipo	Tamaño	Descripción	Tipo de Restricción
Codigo_Ingreso	int	8	Contiene el código de ingreso de los productos.	Clave primaria. No nulo.
Anio_Ingreso	varchar	4	Contiene el año en que se ingresa el pedido.	No nulo.
Codigo_Proveedor	int	4	Contiene el código que identifica al proveedor.	No nulo.
Codigo_Organizacion	int	4	Contiene el código que identifica a la organización.	No nulo.
Fecha_TransaccionI	date		Contiene la fecha en que se realiza la transacción.	No nulo.
Detalle_TransaccionI	text		Describe el detalle de ingreso de la transacción.	No nulo.
Codigo_Funcionaria	int	4	Código que identifica a la funcionaria.	No nulo.
Codigo_TDocumento	int	4	Código que identifica el tipo de documento.	No nulo.
Codigo_TTransaccion	int	4	Código que identifica el tipo de transacción.	No nulo.
<b>Clave primaria</b>	Codigo_DIngreso			
<b>Clave foránea</b>	Codigo_Ingreso			
	Anio_Ingreso			
	Codigo_Producto			
<b>Check</b>	Cantidad > 0			

Tabla17. Estructura para la tabla Ingreso (Ingreso): Los Autores.

<b>TABLA: Objetivo</b> En esta tabla se almacenarán la información correspondiente a datos que serán usados para realizar una respectiva planificación sobre los materiales en el centro de documentación y difusión.				
Nombre	Tipo	Tamaño	Descripción	Tipo de Restricción
Codigo	varchar	12	Código que se le asiga al objetivo del plan.	Pk
Codigo_P	varchar	12	Contiene al código que identifica al objeto.	Fk
Codi- go_Funcionaria	varchar	30	Contiene el código que identifica a la funcionaria.	Fk
Descripcion	int	4	Detalle del objetivo propuesto.	No nulo.
Fecha_I	date		Contiene la fecha Inicial de la meta, objetivo o actividad.	ck
Fecha_F	date		Contiene la fecha final de la meta, objetivo o actividad.	ck
Val_Asig	float	8	Especifica el valor asignado a meta o actividad.	No nulo.
Val_Ref	float	8	Especifica el valor reformado.	No nulo.
Val_Plan	float	8	Especifica el valor planificado a usar.	No nulo.
Val_Comp	float	8	Especifica el valor que se uso realmente.	No nulo.
<b>Clave primaria</b>	Codigo			
<b>Clave foránea</b>	Codigo_P		<b>Tabla de referencia</b>	Objetivo
	Codigo_Funcionaria		<b>Tabla de referencia</b>	Funcionaria
<b>Check</b>	Fecha_I		Tipo fecha en formato dd/mm/aa	
	Fecha_F		Tipo fecha en formato dd/mm/aa y debe ser igual o mayor a Fecha_I	

Tabla18. Estructura para la tabla Objetivo (Objetivo): Los Autores.

TABLA: TipoDocumento				
En esta tabla se almacenarán la información correspondiente a los tipos de documentos que se usan para realizar los respectivos egresos o ingresos.				
Nombre	Tipo	Tamaño	Descripción	Tipo de Restricción
Codigo_TDocumento	int	4	Código generado secuencialmente que identifica el tipo de documento.	pk.
Descripcion_TDocumento	varchar	50	Descripción del tipo de documento.	Ch.
Clave primaria	Codigo_TDocumento			
Clave foránea	Tabla de referencia			
Check	El campo no debe estar vacío			

Tabla19. Estructura para la tabla Tipo de documento (Tipodocumento): Los Autores.



<b>TABLA: TipoProducto</b> En esta tabla se almacenarán la información correspondiente a los diferentes tipos de productos que tienen en el centro de Documentación y Difusión.				
Nombre	Tipo	Tamaño	Descripción	Tipo de Restricción
Codigo_TPProducto	int	4	Código generado secuencialmente que identifica el tipo de producto.	pk.
Descripcion_TPProducto	varchar	50	Descripción del tipo de producto.	ck.
<b>Clave primaria</b>	Codigo_TPProducto			
<b>Clave foránea</b>	Tabla de referencia			
<b>Check</b>	El campo no debe estar vacío			

Tabla 20. Estructura para la tabla Tipo de producto (Tipoproducto): Los Autores.

TABLA: DestinoGeo				
En esta tabla se almacenarán la información correspondiente a la ubicación geográfica que será usada como información de los productos que sean donados o difundidos.				
Nombre	Tipo	Tamaño	Descripción	Tipo de Restricción
Codigo_Ubicacion	int	4	Es el código de ubicación que identifica la situación geográfica.	pk.
Descripcion_Ubicacion	varchar	100	Contiene la ubicación geográfica, a donde se destinaran los productos.	ck.
Clave primaria	Codigo_Ubicacion			
Clave foránea	Tabla de referencia			
Check	El campo no debe estar vacío			

Tabla21. Estructura para la tabla Destino geográfico (Destinogeo): Los Autores.

<b>TABLA: ObjetoTransaccion</b> En esta tabla se almacenarán la información correspondiente al tipo de transacción u objeto de transacción que se realiza, puede ser ingreso de nuevos productos por compra, donación o difusión de materiales para eventos, conferencias, foros, etc.				
Nombre	Tipo	Tamaño	Descripción	Tipo de Restricción
Codigo_TTTransaccion	int	4	Es el código tipo de transacción que identifica a la transacción.	pk.
Descripcion_TTTransaccion	varchar	50	Especifica para qué tipo de evento se destina los productos.	ck.
<b>Clave primaria</b>	Codigo_TTTransaccion			
<b>Clave foránea</b>	Tabla de referencia			
<b>Check</b>	El campo no debe estar vacío			

Tabla22. Estructura para la tabla Objeto de transacción (Objetotransaccion): Los Autores.

TABLA: Organización				
En esta tabla se almacenarán la información correspondiente a las organizaciones que son usuarios del centro de documentación y difusión.				
Nombre	Tipo	Tamaño	Descripción	Tipo de Restricción
Codigo_Organizacion	int	4	Es el código con el que se identifica la organización.	pk.
Nombre_Organizacion	varchar	60	Contiene el nombre de la organización.	ck.
Direccion_Organizacion	varchar	70	Contiene la dirección de la organización.	No nullo.
Telefono_Organizacion	varchar	20	Contiene el teléfono de la organización.	No nullo.
<b>Clave primaria</b>	Codigo_Organizacion			
<b>Clave foránea</b>			<b>Tabla de referencia</b>	
<b>Check</b>	Nombre_Organizacion		El campo no debe estar vacío	

Tabla23. Estructura para la tabla Organización (Organización): Los Autores.

TABLA: Privilegio				
En esta tabla se almacenarán el privilegio según el perfil que tenga cada usuario que este registrado en el sistema.				
Nombre	Tipo	Tamaño	Descripción	Tipo de Restricción
Codigo_Privilegio	varchar	4	Contiene al código que identifica al privilegio.	Pk.
Descripcion_Privilegio	varchar	20	Describe al privilegio que se usara para el acceso al sistema	ck.
Clave primaria	Codigo_Privilegio			
Clave foránea	Tabla de referencia			
Check	El campo no debe estar vacío			

Tabla24. Estructura para la tabla Privilegio (Privilegio): Los Autores.

TABLA: Responsable				
En esta tabla se almacenarán la información personal correspondiente a la persona que recibe los materiales designada por cada organización o usuario final.				
Nombre	Tipo	Tamaño	Descripción	Tipo de Restricción
Codigo_Responsable	int	4	Código generados secuencialmente que identifica al responsable.	pk.
Cedula_Responsable	varchar	10	Contiene la cedula de persona responsable.	ck.
Nombre_Responsable	varchar	25	Contiene el nombre del responsable.	ck.
Telefono_Responsable	varchar	20	Contiene el teléfono del responsable.	No nulo.
<b>Clave primaria</b>	Codigo_Responsable			
<b>Check</b>	Cedula_Responsable		Valor numérico no mayor de 10 dígitos	
	Nombre_Responsable		El campo no debe estar vacío	

Tabla25. Estructura para la tabla Responsable (Responsable): Los Autores.

TABLA: Tipo_Servicio				
En esta tabla se almacenarán la información correspondiente al tipo de servicio que se ofrece para la evaluación que realiza el usuario del centro de documentación y difusión, como puede ser, vía telefónica, personal, email, etc.				
Nombre	Tipo	Tamaño	Descripción	Tipo de Restricción
Codigo_TServicio	int	4	Código generado secuencialmente que identifica al tipo de servicio.	pk.
Descripcion_TServicio	varchar	50	Describe el detalle del servicio entregado.	ck.
Clave primaria	Codigo_TServicio			
Clave foránea			Tabla de referencia	
Check	Descripcion_TServicio		El campo no debe estar vacío	

Tabla26. Estructura para la tabla Tipo de servicio (Tipo\_servicio): Los Autores.

<b>TABLA: Tipo_Usuario</b> En esta tabla se almacenarán la información correspondiente al tipo de usuario que hace uso de los materiales que se encuentran en el centro de documentación y difusión, pueden ser estudiantes de colegio, universitarios, funcionarios públicos, etc.				
Nombre	Tipo	Tamaño	Descripción	Tipo de Restricción
Codigo_TUsuario	int	4	Código generado secuencialmente que identifica al tipo de usuario.	pk.
Descripcion_TUsuario	varchar	50	Descripción del tipo de usuario.	ck.
<b>Clave primaria</b>	Codigo_TUsuario			
<b>Clave foránea</b>	Tabla de referencia			
<b>Check</b>	El campo no debe estar vacío			

Tabla27. Estructura para la tabla Tipo de usuario (Tipo\_usuario): Los Autores.



TABLA: Tipo_Unidad				
En esta tabla se almacenarán la información correspondiente a los diferentes tipos de unidad con los que se presentan o se almacenan los productos, pueden ser, unidad, docenas, medias docenas, pacas, resmas, etc.				
Nombre	Tipo	Tamaño	Descripción	Tipo de Restricción
Codigo_TUnidad	varchar	5	Código de tipo de unidad generado por las dos primeras letras del tipo de unidad seguido de un número secuencial.	pk.
Descripcion_TUnidad	varchar	50	Descripción del tipo de unidad.	fk.
Clave primaria	Codigo_TUnidad			
Clave foránea	Tabla de referencia			
Check	El campo no debe estar vacío			

Tabla28. Estructura para la tabla Tipo de unidad (TipoUnidad): Los Autores.

<p><b>TABLA: Detalle_Egreso</b></p> <p>En esta tabla se almacenarán la información correspondiente al detalle de los productos que se envían sean por donación o difusión.</p>				
Nombre	Tipo	Tamaño	Descripción	Tipo de Restricción
Codigo_DEgreso	int	8	Código secuencial que identifica al detalle de egresos.	pk No nulo.
Codigo_Egreso	int	8	Contiene el código que identifica al egreso de los productos.	fk.
Anio_Egreso	varchar	4	Contiene el código que identifica el año del egreso.	No nulo.
Codigo_Producto	varchar	5	Contiene el código que identifica a los productos.	fk
Cantidad	int	8	Contiene el código que identifica la cantidad del egreso.	Ck.
<b>Clave primaria</b>		Codigo_DEgreso		
<b>Clave foránea</b>		Codigo_Egreso	<b>Tabla de referencia</b>	Egreso
		Codigo_Producto	<b>Tabla de referencia</b>	Producto
<b>Check</b>		Cantidad	Cantidad > 0	

Tabla29. Estructura para la tabla Detalle de egreso (Detalle\_Egreso): Los Autores.

<b>TABLA: Registro_User</b> En esta tabla se almacenarán la información a los datos personales de los usuarios que usan los materiales del centro de documentación y difusión de la institución y que posteriormente realizan la evaluación de cómo han sido atendidos.				
Nombre	Tipo	Tamaño	Descripción	Tipo de Restricción
Codigo_User	int	8	Código generado secuencialmente que identifica al usuario.	pk.
Cedula_User	varchar	10	Contiene la cedula del usuario.	dk.
Nombre_User	varchar	25	Contiene el nombre del usuario.	dk.
Apellido_User	varchar	25	Contiene el apellido del usuario.	dk.
Sexo_User	varchar	10	Contiene el género del usuario.	No nulo.
Institucion_User	varchar	70	Especifica el nombre de la institución.	dk.
Email_User	text		Contiene el correo electrónico del usuario.	No nulo.
Tema_Conсульта	varchar	30	Descripción del tema a consultar.	dk.
Telefono_User	varchar	15	Contiene el teléfono del usuario.	No nulo.
Fecha_Conсульта	date		Contiene la fecha de la consulta realizada.	dk.
Codigo_TUsuario	int	4	Código de tipo de usuario.	fk.
Codigo_TServicio	int	4	Código de tipo de servicio.	fk.
Codigo_Funcionaria	int	4	Código de la funcionaria.	fk.
<b>Clave primaria</b>	Codigo_User			
<b>Clave foránea</b>	Codigo_TUsuario		<b>Tabla de referencia</b>	Tipo_Usuario
	Codigo_TServicio		<b>Tabla de referencia</b>	Tipo_Servicio
	Codigo_Funcionaria		<b>Tabla de referencia</b>	Funcionaria
<b>Check</b>	Cedula_User		Campo numérico de 10 dígitos El campo no debe estar vacío y debe ser de 10 dígitos	
	Nombre_User		El campo no debe estar vacío	
	Apellido_User		El campo no debe estar vacío	
	Institución_User		El campo no debe estar vacío	
	Tema_Conсульта		El campo no debe estar vacío	
	Fecha_Conсульта		Formato dd/mm/aa, es automático	

Tabla 30. Estructura para la tabla Registro de usuario (Registro\_user): Los Autores.

## 2.2.4. DIAGRAMAS DE SECUENCIA

Un diagrama de secuencia muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo, el diagrama de secuencia contiene detalles de implementación del escenario y mensajes intercambiados entre los objetos.

Para visualizar los diagramas de secuencia dirigirse al Capítulo 6 página 106-111.

## 2.3. MODELO DE DISEÑO

### 2.3.1. DISEÑO DE OBJETOS

En el diseño del sistema se puede identificar la siguiente estructura de objetos:

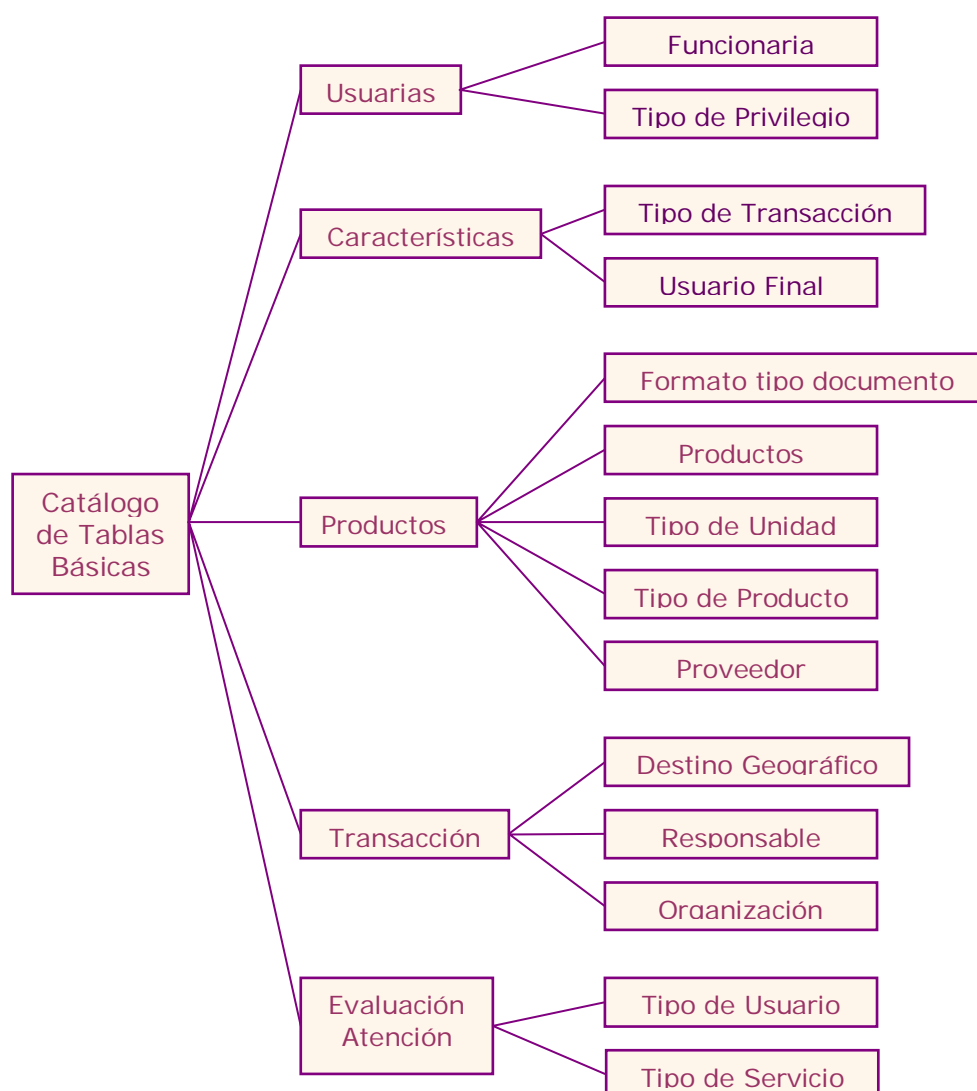


Figura19. Diseño de objetos: Catálogos de tablas básicas; Los Autores.

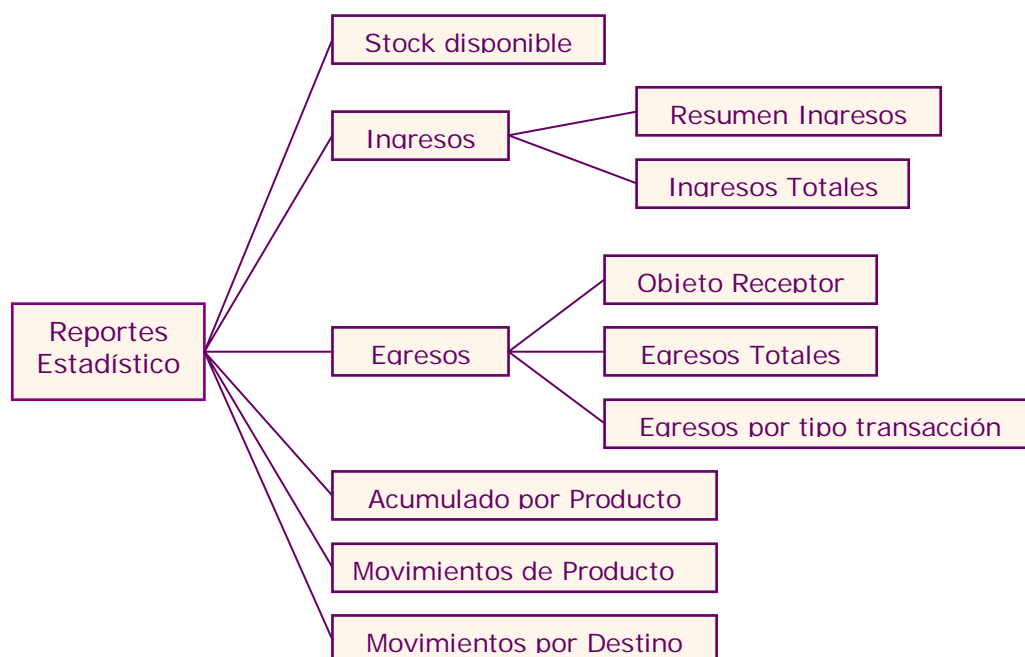


Figura20. Diseño de objetos: Reportes Estadísticos: Los Autores.

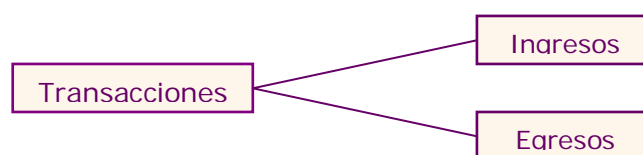


Figura21. Diseño de objetos: Transacciones: Los Autores.

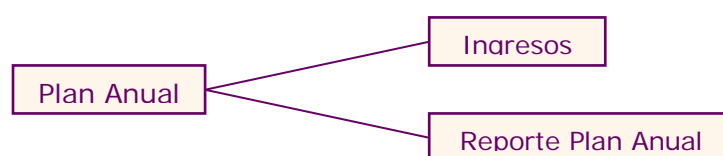


Figura22. Diseño de objetos: Plan Anual: Los Autores.

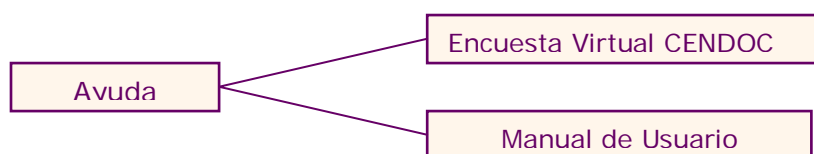


Figura23. Diseño de objetos: Ayuda: Los Autores.

### 2.3.2. DISEÑO DE INTERFACES

En el diseño de interfaces se describe las pantallas que forman parte del sistema; para revisar la interfaz del sistema de Inventario dirigirse al Manual de usuario.

### 2.3.3. DISEÑO DEL SISTEMA

Aquí se define una subdivisión en aplicaciones del sistema y la forma de comunicación con los sistemas existentes con los que debe interactuar.

#### 2.3.4.1. Archivos o Bases de Datos (Diagrama entidad relación)

##### DISEÑO LÓGICO

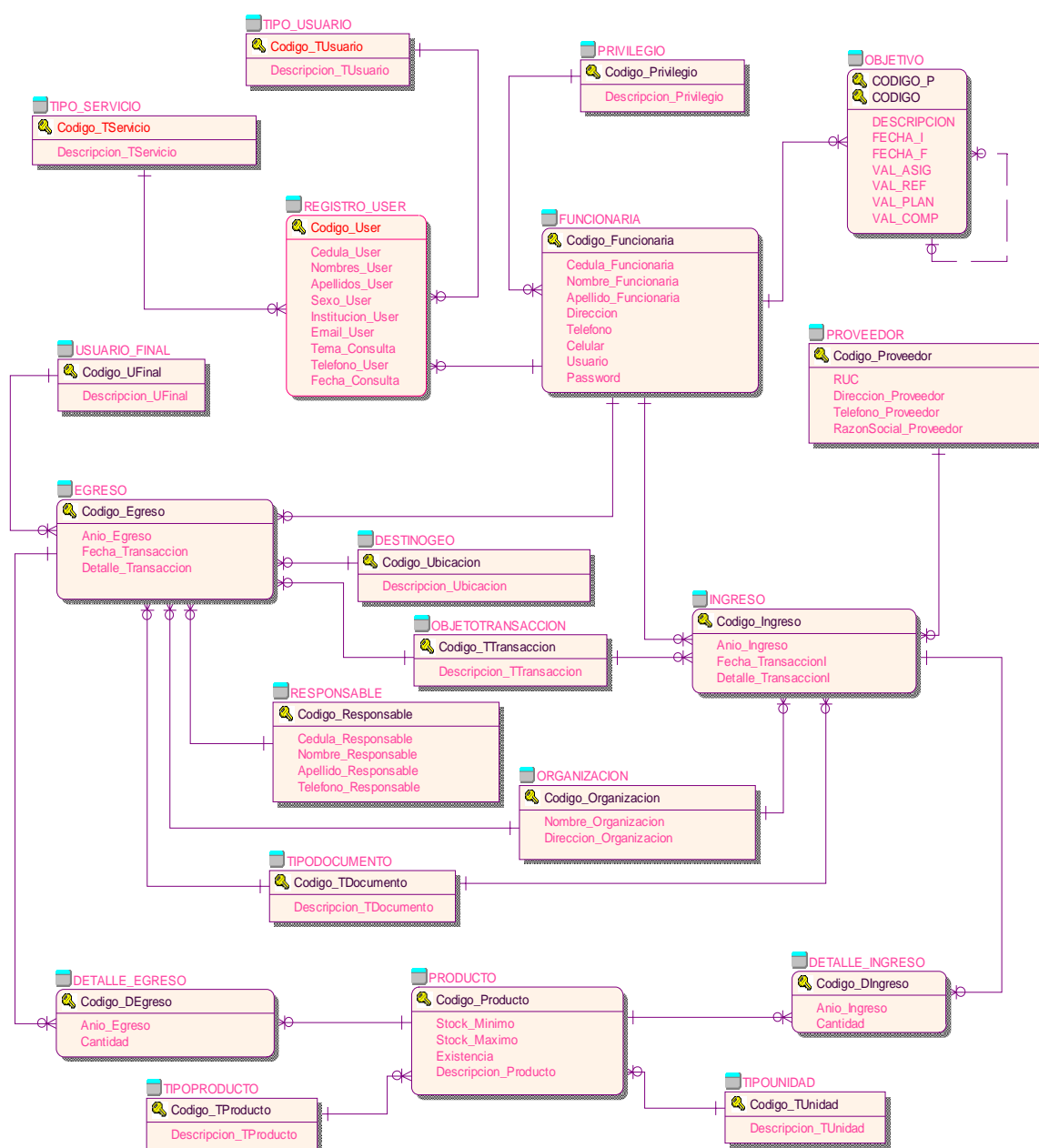


Figura24. Base de Datos: Diseño Lógico: Los Autores.

## DISEÑO FÍSICO

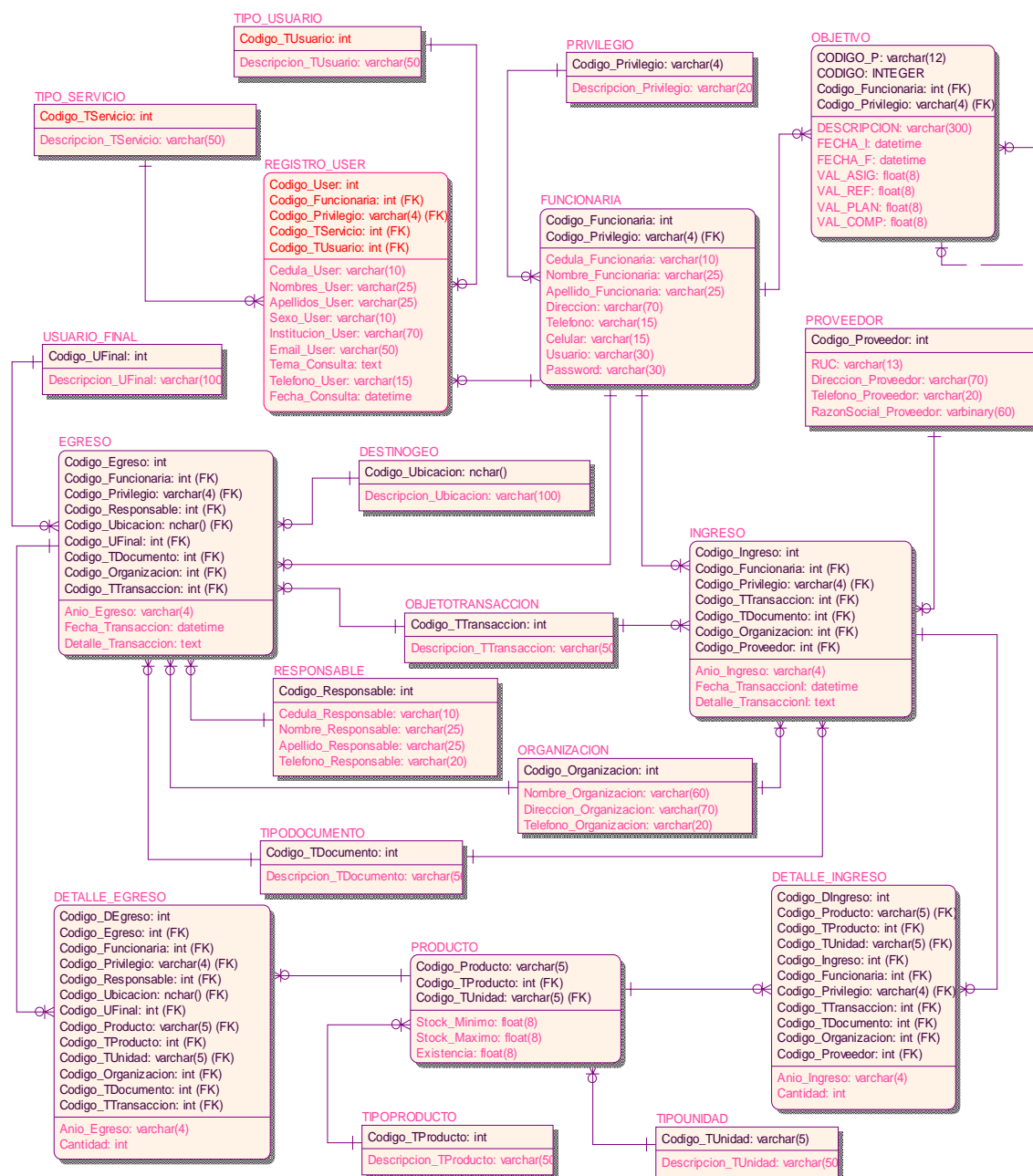


Figura25. Base de Datos: Diseño Físico: Los Autores.

### 2.3.4. DISEÑO DE MENÚS Y MAPAS DE NAVEGACIÓN

A continuación se muestra el diseño de menús donde se indica los ítems que este contiene. También se muestra los mapas de navegación a fin de facilitar el acceso a los diversos contenidos y niveles de su estructura.

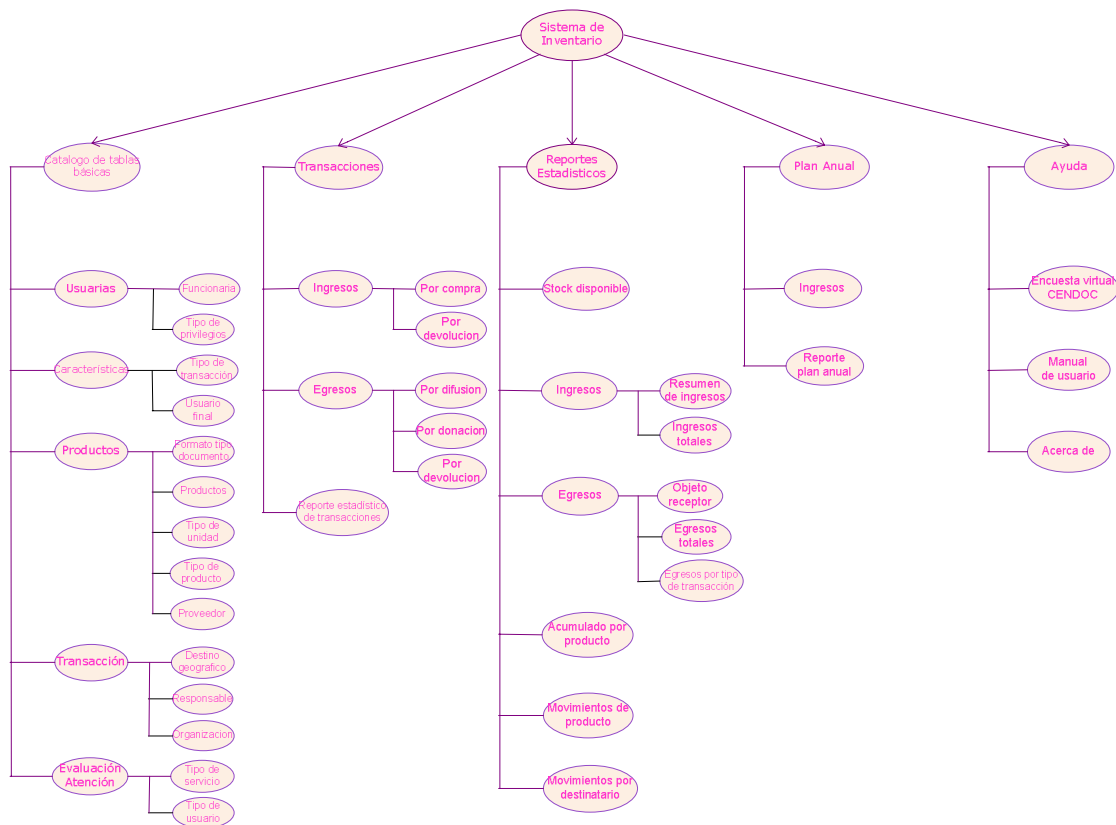


Figura26. Diseño de Menú: Los Autores.

## MAPAS DE NAVEGACIÓN

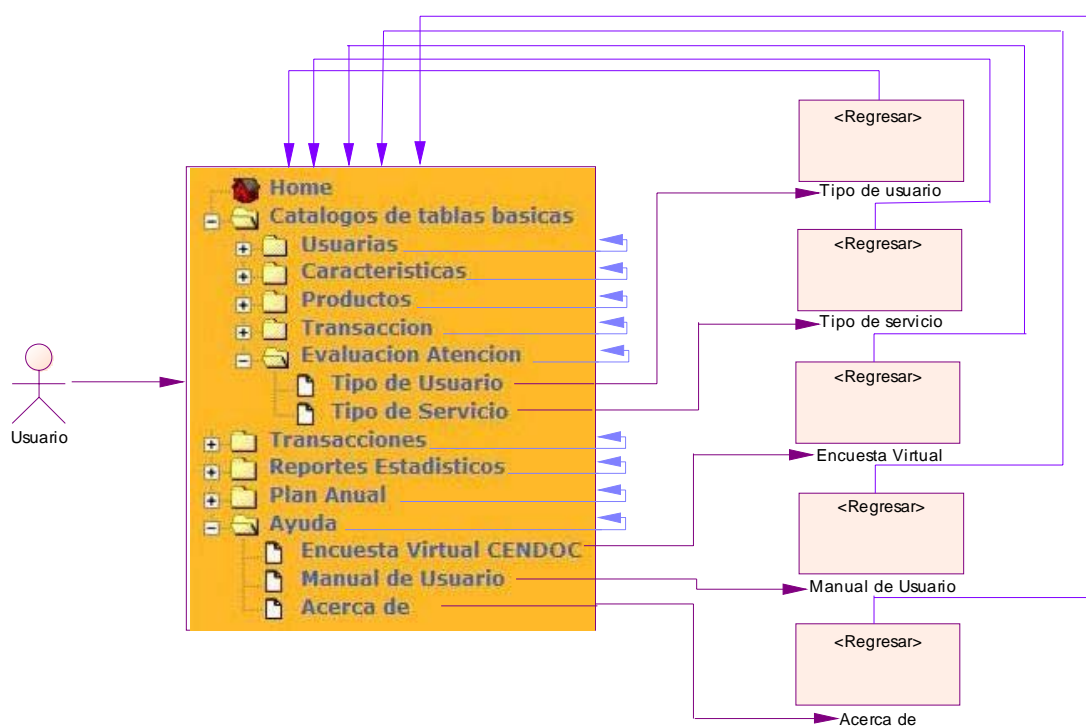


Figura27. Mapa de Navegación: Los Autores.



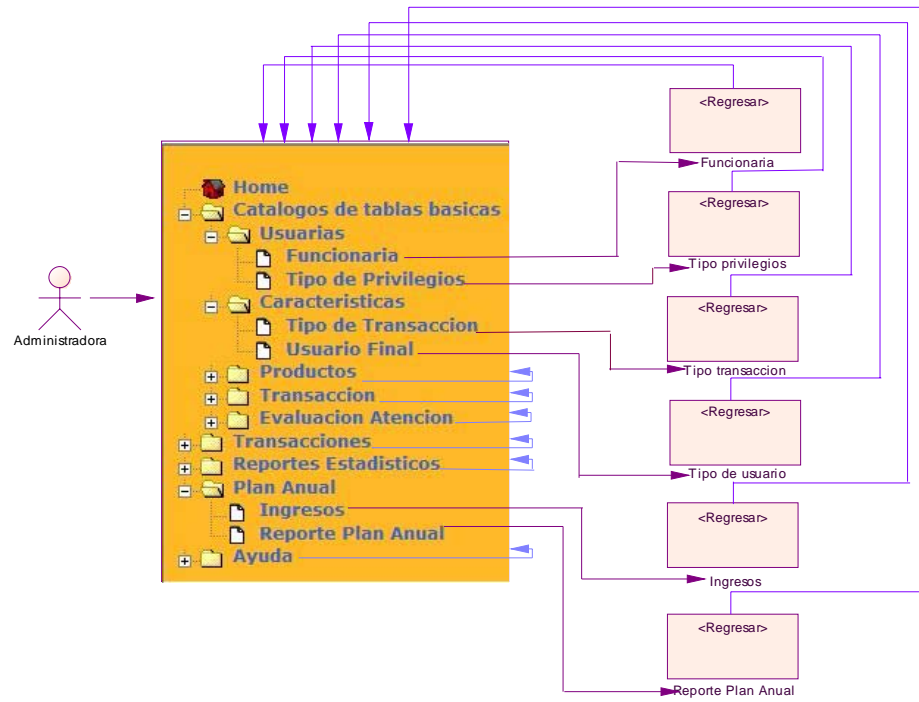


Figura26. Mapa de Navegación: Los Autores.

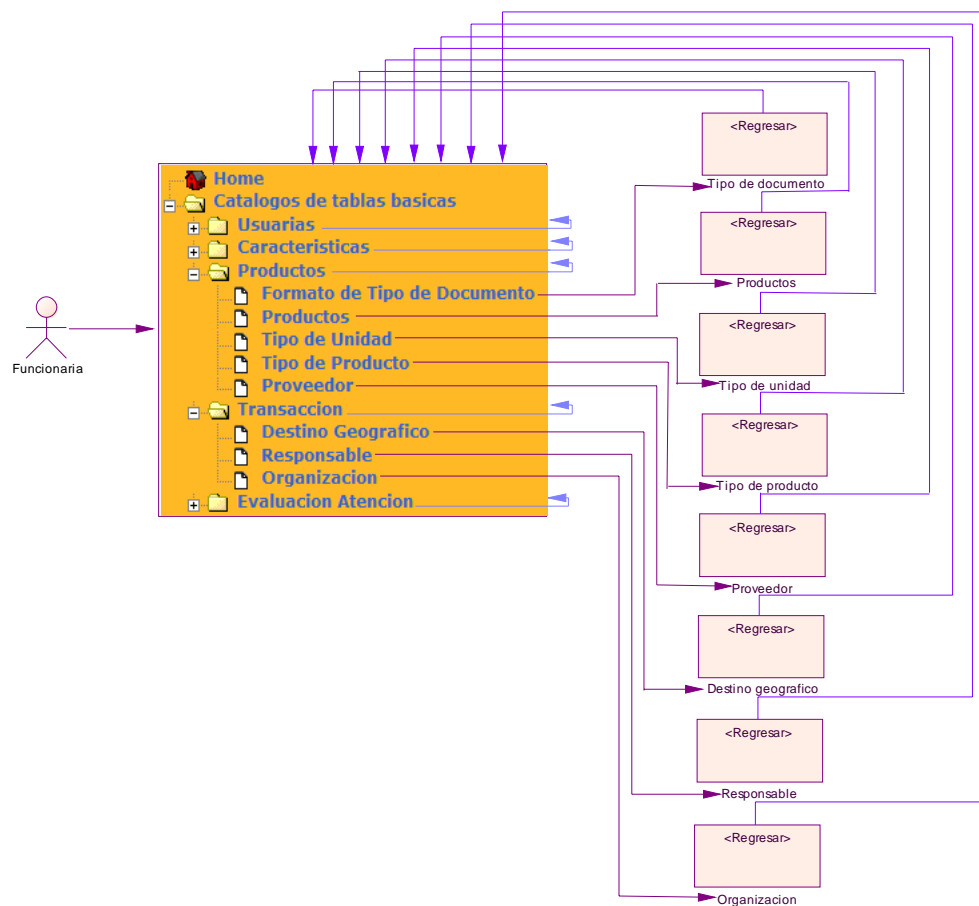


Figura27. Mapa de Navegación: Los Autores.

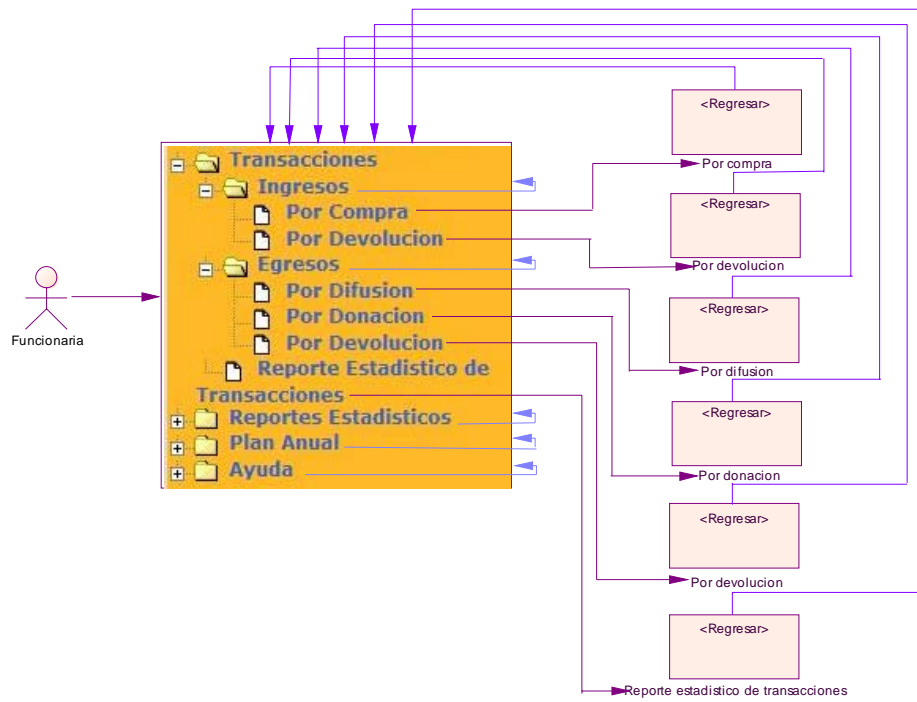


Figura28. Mapa de Navegación: Los Autores.

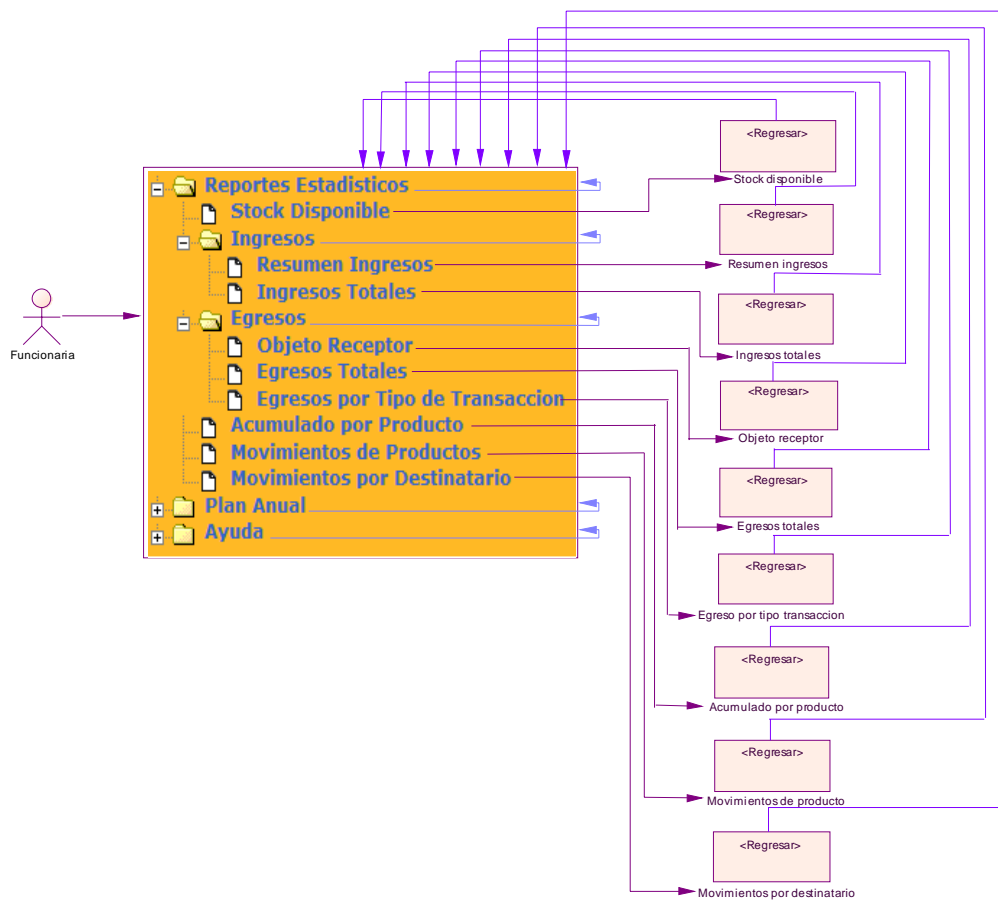


Figura29. Mapa de Navegación: Los Autores.

### 2.3.5. DISEÑO DE ARQUITECTURA DEL SISTEMA

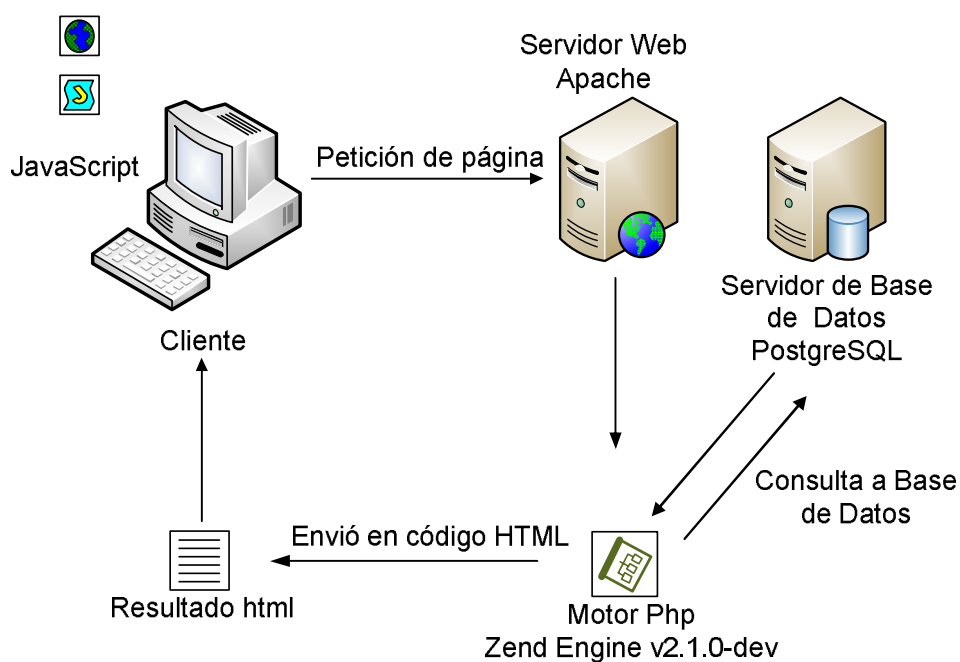


Figura30. Arquitectura del Sistema<sup>2</sup>: Los Autores.

## 3. DIAGRAMA GLOBAL DEL SISTEMA

### 3.1. DIAGRAMA DE COMPONENTES

Diagrama de componentes que permite mostrar una vista física del modelo como también los componentes de software y las relaciones que existen entre ellos<sup>3</sup>.

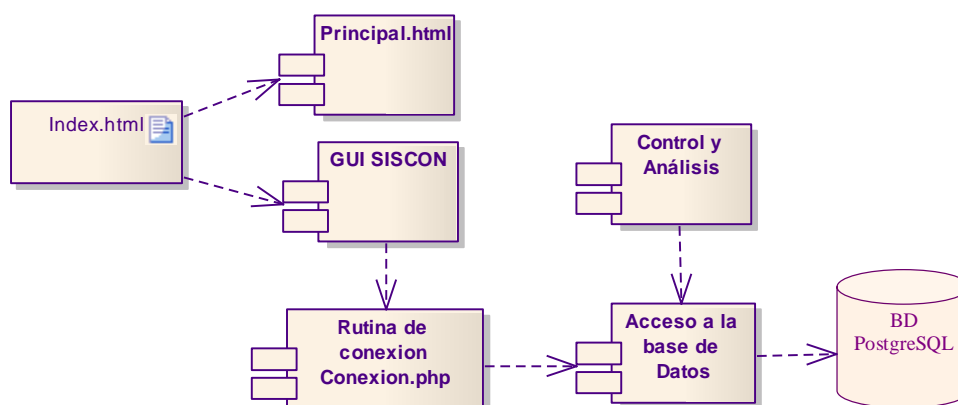


Figura31. Diagrama de componente: Los Autores.

<sup>2</sup> Versión del motor de lenguaje de Scripting de Zend que hace uso el programa.

<sup>3</sup> UML con Rational Rose – Ing. Gesvin Romero Moreno, Capítulo 1, Pág. 28

### 3.2. DIAGRAMA DE DISTRIBUCIÓN

Los diagramas de distribución se enfocan en la parte de hardware, cada uno de los nodos puede contener otros componentes, incluyendo software.

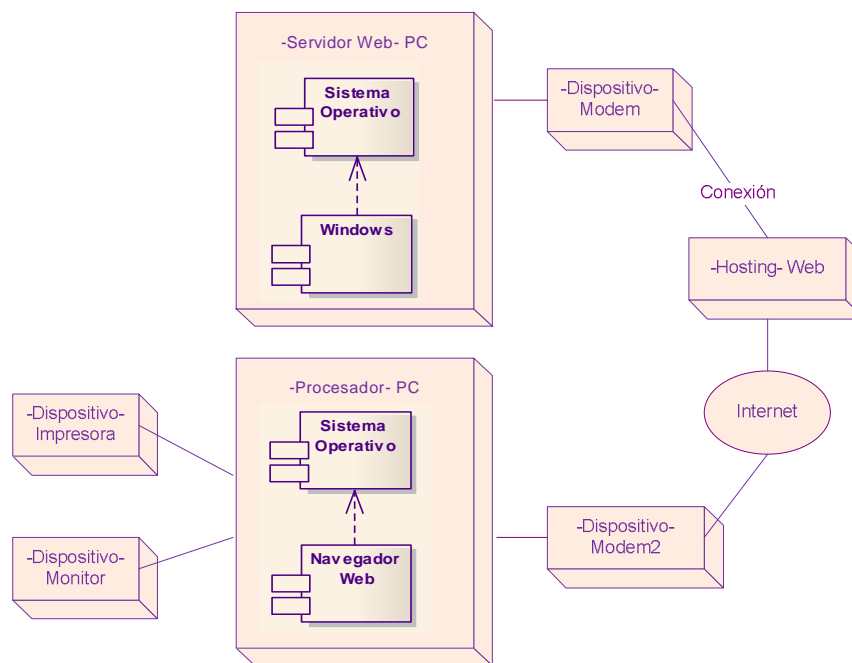


Figura32. Diagrama de Distribución: Los Autores.

## 4. INSTRUCCIONES DE INSTALACIÓN Y EJECUCIÓN

### 4.1. REQUERIMIENTOS DE HARDWARE Y SOFTWARE

#### Para Servidor:

- Sistema Operativo Linux (puede ser cualquier plataforma)
- Sistema Operativo Windows 2000 Server
- Servidor Web – Apache 4.5
- Servidor de Base de Datos – PostgreSQL 8.4.0

#### Requisitos mínimos del sistema

- Con procesador Intel Pentium IV a 1.8 GHz o superior
- 512 MB de memoria RAM
- 8 GB de espacio de disco disponible
- Pantalla con resolución 800 x 600

#### Configuración recomendada

- PC con procesador Intel Pentium IV 2.4 GHz

- 1 GB de memoria RAM
- 40 GB de espacio de disco disponible
- Pantalla con resolución 1024 x 768

**Para Terminal:**

- Sistema Operativo Windows 98 o superior
- Navegador de Internet: Internet Explorer, Netscape Navigator, Firefox Mozilla.

**Requisitos mínimos del sistema**

- PC con procesador Intel Pentium IV de 1.8 Ghz o superior
- 256 Mb de memoria RAM
- 4 GB de espacio de disco disponible
- Pantalla con resolución 800 x 600 o superior

**RED**

- Red LAN. Por la facilidad que se puede realizar cambios especialmente en el hardware.
- Capacidad de Transmisión de 100Mbps.
- Medio de Transmisión: Cable UTP Categoría 5 o superior.
- Topología de Red: la que este en uso en la institución.

## **5. SEGURIDAD Y CONTROL**

### **5.1 SEGURIDAD EN EL ACCESO DE LA INFORMACIÓN**

Una vez ingresado el nombre de usuario y password el sistema hará una consulta a la base de datos autenticando la información ingresada, caso contrario arrojará un mensaje de error.

Si los datos son correctos el sistema verificará a través de la base de datos el perfil que se le haya asignado al usuario, de esta manera cuando el usuario ingrese al sistema según su perfil las opciones del menú estarán habilitadas o deshabilitadas.

### **5.2 CREACIÓN DE USUARIOS**

La creación de usuarios será potestad del administrador de sistema, él cual tendrá que ingresar información personal de futuro usuario como lo son sus nombres,

apellidos dirección, etc., además cada uno de ellos tendrán un identificador que será la primera letra de su nombre seguido por su apellido completo y una clave que le permitirá acceder a su cuenta en el sistema y poder acceder a las diversas actividades que tiene permitido que fueron asignadas por el administrador cuando la cuenta fue creada.

### **5.3 SEGURIDAD DE ACCESO**

Si se desea realizar alguna tarea se deberá tener una cuenta que permita realizarlo y acceder desde el enlace que solicita el usuario y la contraseña de la cuenta a la cual se desea ingresar para operar en el sistema y realizar las actividades que están descritas en el rol de usuario; caso contrario no se puede acceder al sistema en ninguna área, ni realizar actividad alguna que pueda modificar la información que se registra en la base de datos.

### **5.4 SEGURIDAD DE USUARIO**

Todos los usuarios tienen que tener especial seguridad con la contraseña de su cuenta dentro del sistema, porque todos los cambios que se realicen dentro de la misma son responsabilidad del propietario de dicha cuenta.

Por otra parte a cada uno de los usuarios le es asignado un rol, estos determinan las actividades que pueden o no realizar cada tipo de usuario.

Los roles están divididos de la siguiente manera:

- **Administrador:** Puede realizar todas las operaciones y controles que el sistema permite realizar, así como realizar modificaciones y operaciones directamente en la base de datos.
- **Funcionaria:** Este rol permite realizar modificaciones en cuanto a la información que será almacenada en determinadas tablas como lo son: la planificación de actividades, registro de material, encuestas, etc.
- **Usuario:** Este rol está ligado a los beneficiarios como son las instituciones y organizaciones. Conjuntamente de los usuarios que realicen la encuesta virtual del centro de documentación.

## 5.5 COPIAS DE SEGURIDAD

Dependiendo del plan de contingencia de la institución, de la base de datos del sistema se deberá obtener una copia de seguridad cada intervalo de tiempo el cual es designado por la institución (sea semanal, mensual, trimestral, anual, etc.) esta copia deberá ser almacenada en el disco del servidor para luego ser recopilada en un dispositivo externo (sea CD, DVD, discos externos), etiquetando y especificando los intervalos de las fechas en las cuales fueron obtenidas.

## 5.6 LISTA DE MENSAJES DE ERROR

- `echo "<div align='center'><FONT size = '3' face = 'verdana'>DEBE  
ESPECIFICAR UN USUARIO</FONT></div>";`
- `echo "<div align='center'><FONT size = '3' face = 'verdana'>USUARIO NO  
EXISTENTE, POR FAVOR COMUNIQUESE CON  
INFORMÁTICA</FONT></div>";`
- `echo "<p align='center' class='TextoTabla1'>Funcionaria ya existe: Datos no  
se grabaron</p>";`
- `echo "<p align='center' class='TextoTabla1'>Tipo Transaccion ya existe: Datos  
no se grabaron</p>";`
- `echo "<p align='center' class='TextoTabla1'>Usuario final ya existe: Datos no  
se grabaron</p>";`
- `echo "<p align='center' class='TextoTabla1'>Tipo de documento ya existe:  
Datos no se grabaron</p>";`
- `echo "<p align='center' class='TextoTabla1'>Producto ya existe: Datos no se  
grabaron</p>";`
- `echo "<p align='center' class='TextoTabla1'>Tipo de unidad ya existe: Datos  
no se grabaron</p>";`
- `echo "<p align='center' class='TextoTabla1'>Tipo de producto ya existe: Datos  
no se grabaron</p>";`
- `echo "<p align='center' class='TextoTabla1'>Destino Geografico ya existe:  
Datos no se grabaron</p>";`
- `echo "<p align='center' class='TextoTabla1'>Organizacion ya existe: Datos no  
se grabaron</p>";`

- echo "<p align='center' class='TextoTabla1'>Tipo de Usuario ya existe: Datos no se grabaron</p>";
- echo "<p align='center' class='TextoTabla1'>Tipo de Servicio ya existe: Datos no se grabaron</p>";
- echo "<TD width=541 bgColor=#9a4d9f><div align='center'><font face = 'Verdana' color = 'white' size = '3'> <strong>DEBE INICIAR LA SESION CON UNA CUENTA EXISTENTE</strong></font></div></TD>";

## 6. CÓDIGO FUENTE

### Index.html

```
<html>
<head>
<title>SISCON - Ingreso</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<head><body><table width="551" border="0" align="center" cellpadding="0" cellspacing="0"
background="Imagenes/Fondo.jpg">
<tr> <td height="244" valign="bottom">&nbsp;</td></tr>
<tr> <td width="551" height="125" valign="middle">
<form name="form1" method="post" action="Validar_Usuario.php">
<table width="286" border="0" align="center" cellpadding="0" cellspacing="0" bordercolor="#000000">
<tr> <td width="108" height="27" class="Estilo3"><div align="right">Usuario:</div></td>
<td width="8">&nbsp;</td>
<td width="148"><input name="txt_User" type="text" id="txt_User" size="25" maxlength="20"></td>
</tr>
<tr> <td height="31" class="Estilo3"><div align="right">Contrase&ntilde;a:</div></td>
<td>&nbsp;</td>
<td><input name="txt_Password" type="password" id="txt_Password" size="25"
maxlength="20"></td></tr>
</table>
<table width="200" height="32" border="0" align="center" cellpadding="0" cellspacing="0">
<tr> <td><div align="center">
<input name="btn_Aceptar" type="submit" id="btn_Aceptar" value="Aceptar">
</div></td>
<td><div align="center"></div></td>
<td><div align="center">
<input name="btn_Cancelar" type="reset" id="btn_Cancelar" value="Cancelar">
</div></td>
</tr> </table>
</form></td></tr></table>
</body>
```

### Validar\_Usuario.php

```
<?php
session_start();
//Datos de la BD
$user='postgres';
$password='123';
$host='localhost';
$db_name='SICCDD';
include 'Conexion.php';
function quitar($mensaje){
    // almacenamiento en variables
    $usuario = $_POST[txt_User];
    $password = $_POST[txt_Password];
    $nopermitidos = array("'",'\'','<','>','\"');
```



```

    $mensaje = str_replace($nopermitidos, "", $mensaje);
    return $mensaje;
}
if(trim($_POST[txt_User]) != ""){
    $usuario = $_POST[txt_User];
    $password = $_POST[txt_Password];
    $result = pg_exec("SELECT codigo_funcionaria, nombre_funcionaria, usuario, password
    FROM funcionaria
    WHERE usuario like '%".$usuario. "%' and password like '%".$password. "%'");
    $filas=pg_numrows($result);
    if($filas != 0){
        $usuario1 = pg_result($result, 0, 0);
        $_SESSION["k_username"] = $usuario1;
        echo 'Has sido logueado correctamente ' .$_SESSION['k_username']. ' <p>';
        echo '<a href="Principal.php">Index</a></p>';
        //Elimina el siguiente comentario si quieres que re-dirigir automáticamente a index.php
        /*Ingreso exitoso, ahora sera dirigido a la pagina principal.*/
        ?>
        <SCRIPT LANGUAGE="javascript"> location.href = "Principal.html";
        </SCRIPT>
        <?php
        }

    else{
        echo "<div align='center'><FONT size = '3' face = 'verdana'>USUARIO NO EXISTENTE, POR
        FAVOR COMUNIQUESE CON INFORMÁTICA</FONT></div>";
    }
    pg_FreeResult($result);
}
else{
    echo "<div align='center'><FONT size = '3' face = 'verdana'>DEBE ESPECIFICAR UN
    USUARIO</FONT></div>";
}
?>

```

#### Principal.html

```

<html>
<head>
<title>SISCON</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<frameset rows="123,*" cols="" frameborder="NO" border="0" framespacing="0">
    <frame src="Cabecera.html" name="topFrame" scrolling="NO" noresize>
    <frameset rows="*,36" cols="245,211,*,231" framespacing="0" frameborder="NO" border="0">
        <frame src="vacio.html">
        <frame src="Menu.html" name="leftFrame" scrolling="NO" noresize>
        <frame src="vacio.html" name="framep">
        <frame src="vacio.html">
    </frameset>
</frameset>
<noframes><body>
</body></noframes>
</html>

```

#### Menu.html

```

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>SISCON</title>
<script language="JavaScript" type="text/javascript">
/*****
Menú en arbol.
*****/
var anMenu = 300
var totalMen =6

```

```

var anlmas = 17
var allmas = 15
var direc = 'iconos'
var mas = '/mast.gif'
var menos = '/menost.gif'
var puntos = '/puntost.gif'
var puntosv = '/puntosvt.gif'
var carpeab = '/carpabiertat.gif'
var carpece = '/carpcerradat.gif'
var puntosu = '/puntosut.gif'
var doc = '/doct.gif'
var docsel = '/docsel.gif'
var carpeabsel = '/carpabiertasel.gif'
var carpecesel = '/carpcerradasel.gif'
var icHome = '/home.gif'
var puntosh = '/puntosh.gif'
function tunMen(tex,enl,dest,subOp,an){
this.tex = tex;
this.enl = enl;
this.dest = dest;
this.subOp = subOp;
this.an = an;
this.secAc = false
}
var Op_0 = new tunMen("Home",null,null,0)
var Op_1 = new tunMen("Catalogos de tablas basicas",null,null,5)
var Op_1_0=new tunMen('Usuarías',null,null,2)
var Op_1_0_0=new tunMen('Funcionaria','Funcionaria.php','framep',0)
var Op_1_0_1=new tunMen('Tipo de Privilegios','Privilegio.html','framep',0)
var Op_1_1=new tunMen('Características',null,null,2)
var Op_1_1_0=new tunMen('Tipo de Transaccion','ObjetoTransaccion.html','framep',0)
var Op_1_1_1=new tunMen('Usuario Final','UsuarioFinal.html','framep',0)
var Op_1_2=new tunMen('Productos',null,null,5)
var Op_1_2_0=new tunMen('Formato de Tipo de Documento','TipoDocumento.html','framep',0)
var Op_1_2_1=new tunMen('Productos','Productos.php','framep',0)
var Op_1_2_2=new tunMen('Tipo de Unidad','TipoUnidad.html','framep',0)
var Op_1_2_3=new tunMen('Tipo de Producto','TipoProducto.html','framep',0)
var Op_1_2_4=new tunMen('Proveedor','Proveedores.html','framep',0)
var Op_1_3=new tunMen('Transaccion',null,null,3)
var Op_1_3_0=new tunMen('Destino Geografico','DestinoGeografico.html','framep',0)
var Op_1_3_1=new tunMen('Responsable','Responsable.html','framep',0)
var Op_1_3_2=new tunMen('Organizacion','Organizacion.html','framep',0)
var Op_1_4=new tunMen('Evaluacion Atencion',null,null,2)
var Op_1_4_0=new tunMen('Tipo de Usuario','TipoUsuario.html','framep',0)
var Op_1_4_1=new tunMen('Tipo de Servicio','TipoServicio.html','framep',0)
var Op_2 = new tunMen("Transacciones",null,null,3)
var Op_2_0=new tunMen('Ingresos',null,null,1)
var Op_2_0_0=new tunMen('Generar nuevo Ingreso','Ingresos.php','framep',0)
var Op_2_1=new tunMen('Egresos',null,null,1)
var Op_2_1_0=new tunMen('Generar nuevo Egreso','Egresos.php','framep',0)
var Op_2_2=new tunMen('Reporte Estadístico de Transacciones',null,"",0)
var Op_3 = new tunMen("Reportes Estadísticos",null,null,6)
var Op_3_0=new tunMen('Stock Disponible',null,"",0)
var Op_3_1=new tunMen('Ingresos',null,null,2)
var Op_3_1_0=new tunMen('Resumen Ingresos',null,"",0)
var Op_3_1_1=new tunMen('Ingresos Totales',null,"",0)
var Op_3_2=new tunMen('Egresos',null,null,3)
var Op_3_2_0=new tunMen('Objeto Receptor',null,"",0)
var Op_3_2_1=new tunMen('Egresos Totales',null,"",0)
var Op_3_2_2=new tunMen('Egresos por Tipo de Transaccion',null,"",0)
var Op_3_3=new tunMen('Acumulado por Producto',null,"",0)
var Op_3_4=new tunMen('Movimientos de Productos',null,"",0)
var Op_3_5=new tunMen('Movimientos por Destinatario',null,"",0)
var Op_4 = new tunMen("Plan Anual",null,null,2)

```

```

var Op_4_0=new tunMen('Ingresos',null,"",0)
var Op_4_1=new tunMen('Reporte Plan Anual',null,"",0)
var Op_5 = new tunMen("Ayuda",null,null,3)
var Op_5_0=new tunMen('Encuesta Virtual CENDOC','Evaluacion.php','_blank',0)
var Op_5_1=new tunMen('Manual de Usuario',null,"",0)
var Op_5_2=new tunMen('Acerca de',null,"",0)

var anchoTotal = 1520;

var tunlex=navigator.appName=="Microsoft Internet Explorer"?true:false;
if(tunlex && navigator.userAgent.indexOf('Opera')>=0){tunlex = false}
var manita = tunlex ? 'hand' : 'pointer'
var subOps = new Array()
function construye(){
  cajaMenu = document.createElement('div')
  cajaMenu.style.width = anMenu + "px"
  document.getElementById('tunMe').appendChild(cajaMenu)
  for(m=0; m < totalMen; m++){
    opchon = eval('Op_'+m)
    ultimo = false
    try{
      eval('Op_' + (m+1))
    }
    catch(error){
      ultimo = true
    }
    boton = document.createElement('div')
    boton.style.position = 'relative'
    boton.className = 'botones'
    boton.style.paddingLeft= 0
    carp = document.createElement('img')
    carp.style.marginRight = 5 + 'px'
    carp.style.verticalAlign = 'middle'
    carp2 = document.createElement('img')
    carp2.style.verticalAlign = 'middle'
    enla = document.createElement('a')
    if(opchon.subOp > 0){
      carp2.style.cursor = manita
      carp2.src = direc + mas
      boton.secAc = opchon.secAc
    }
    else{
      carp2.style.cursor = 'default'
      enla.className = 'enls'
      if(ultimo){carp2.src = direc + puntosu}
      else{carp2.src = direc + puntos}
    }
    if(m == 0){
      carp.src = direc + icHome
      carp2.src = direc + puntosh
    }
    else{
      carp.src = direc + carpece
    }
    boton.appendChild(carp2)
    boton.appendChild(carp)
    enla.className = 'enls'
    enla.style.cursor = manita
    boton.appendChild(enla)
    enla.appendChild(document.createTextNode(opchon.tex))
    if(tunlex){
      enla.onmouseover = function(){this.className = 'botonesHover'}
      enla.onmouseout = function(){this.className = 'enls'}
    }
  }
}

```

```

        if(opchon.enl != null && opchon.subOp == 0){
            enla.href = opchon.enl
        }
        if(opchon.dest != null && opchon.subOp == 0){
            enla.target = opchon.dest;
        }
        boton.id = 'op_' + m

        cajaMenu.appendChild(boton)
        if(opchon.subOp > 0 ){
            carp2.onclick= function(){
                abre(this.parentNode,this,this.nextSibling)
            }
            subOps[subOps.length] = boton.id.replace(/o/, "O")
            enla.onclick = function(){
                abre(this.parentNode,this.parentNode.firstChild,this.previousSibling)
            }
        }
    }
    if(subOps.length > 0){subMes()}
}
function subMes(){
    lar = subOps.length
    for(t=0;t<subOps.length;t++){
        opc =eval(subOps[t])
        for(v=0;v<opc.subOp;v++){
            if(eval(subOps[t] + "_" + v + ".subOp") > 0){
                subOps[subOps.length] = subOps[t] + "_" + v
            }
        }
    }
}
construyeSub()
}
var fondo = true
function construyeSub(){
    for(y=0; y<subOps.length;y++){
        opchon = eval(subOps[y])
        capa = document.createElement('div')
        capa.className = 'subMe'
        capa.style.position = 'relative'
        capa.style.display = 'none'
        if(!fondo){capa.style.backgroundImage = 'none'}
        document.getElementById(subOps[y].toLowerCase()).appendChild(capa)
        for(s=0;s < opchon.subOp; s++){
            sopchon = eval(subOps[y] + "_" + s)
            ultimo = false
            try{
                eval(subOps[y] + "_" + (s+1))
            }
            catch(error){
                ultimo = true
            }
            if(ultimo && sopchon.subOp > 0){
                fondo = false
            }
            opc = document.createElement('div')
            opc.className = 'botones'
            opc.id = subOps[y].toLowerCase() + "_" + s
            if(tunlex){
            }
            enla = document.createElement('a')
            enla.className = 'enls'
            enla.style.cursor = manita
            if(sopchon.enl != null && sopchon.subOp == 0){

```

```

        enla.href = sopchon.enl
        if(sopchon.dest != null && sopchon.subOp == 0){
            enla.target = sopchon.dest
        }
    }

    enla.appendChild(document.createTextNode(sopchon.tex))
    capa.appendChild(opc)
    carp = document.createElement('img')
    carp.src = direc + carpece
    carp.style.verticalAlign = 'middle'
    carp.style.marginRight = 5 + 'px'
    carp2 = document.createElement('img')
    carp2.style.verticalAlign = 'middle'
    if(sopchon.subOp > 0){
        opc.secAc = sopchon.secAc
        carp2.style.cursor = manita
        carp2.src = direc + mas
        enla.onclick = function(){
            abre(this.parentNode,this.parentNode.firstChild,this.previousSibling)
        }
        carp2.onclick= function(){
            abre(this.parentNode,this,this.nextSibling)
        }
        if(tunlex){
            enla.onmouseover = function(){this.className = 'botonesHover'}
            enla.onmouseout = function(){this.className = 'enls'}
        }
    }
    else{
        carp2.style.cursor = 'default'
        carp.src = direc + doc
        if(ultimo){carp2.src = direc + puntosu;
        if(sopchon.subOp > 0){alert('hola');capa.style.backgroundImage = 'none'}
        }
        else{carp2.src = direc + puntos}
    }
    opc.appendChild(carp2)
    opc.appendChild(carp)
    opc.appendChild(enla)
}
}
Seccion()
}
function abre(cual,im,car){
    abierta = cual.lastChild.style.display != 'none'? true:false;
    if(abierta){
        cual.lastChild.style.display = 'none'
        im.src = direc + mas
        if(cual.secAc){
            car.src = direc + carpecesel
        }
        else{car.src = direc + carpece}
    }
    else{
        cual.lastChild.style.display = 'block'
        im.src = direc + menos
        if(cual.secAc){car.src = direc + carpeabsel}
        else{car.src = direc + carpeab}
    }
}
var seccion = null
function Seccion(){
    if (seccion != null){

```

```

        if(seccion.length == 4){
document.getElementById(seccion.toLowerCase()).firstChild.nextSibling.src = direc + carpeabsel
document.getElementById(seccion.toLowerCase()).lastChild.className = 'secac2'
document.getElementById(seccion.toLowerCase()).lastChild.onmouseover = function(){
    this.className = 'enls'
}
document.getElementById(seccion.toLowerCase()).lastChild.onmouseout = function(){
    this.className = 'secac2'
}
        }
        else{
document.getElementById(seccion.toLowerCase()).firstChild.nextSibling.src = direc + docsel
document.getElementById(seccion.toLowerCase()).firstChild.nextSibling.nextSibling.className = 'secac'
document.getElementById(seccion.toLowerCase()).parentNode.parentNode.lastChild.previousSibling.className = 'secac2'
document.getElementById(seccion.toLowerCase()).parentNode.parentNode.lastChild.previousSibling.onmouseout = function(){
    this.className = 'secac2'
}
if(!tunlex){
document.getElementById(seccion.toLowerCase()).parentNode.parentNode.lastChild.previousSibling.onmouseover = function(){
    this.className = 'enls'
}
}
document.getElementById(seccion.toLowerCase()).parentNode.parentNode.secAc = true
seccion = seccion.substring(0,seccion.length - 2)
seccionb = document.getElementById(seccion.toLowerCase())
abre(seccionb,seccionb.firstChild,seccionb.firstChild.nextSibling)
if(seccion.length > 4){
lar = seccion.length
for(x = lar; x > 4; x-=2){
    seccion = seccion.substring(0,seccion.length - 2)
    seccionb = document.getElementById(seccion.toLowerCase())
    abre(seccionb,seccionb.firstChild,seccionb.firstChild.nextSibling)
}
}
}
}
onload = construye
</script>
<style type="text/css">
<!--
a.enls:link, a.enls:visited{
color: #3366CC;
text-decoration: none;
}
a.enls:hover{
color: #CC0000;
background-color: #eeeeee;
}
a.secac2{
color: #B87070;
text-decoration: none;
}
a.secac{
color: #FFFFFF;
text-decoration: none;
background-color: #CC0000;
}
a.secac:hover{
color: #B87070;
text-decoration: none;
}

```

```

        background-color: #ffffff;
    }
    .botones {
        font-family: Verdana, Arial, Helvetica, sans-serif;
        font-size: 12px;
        font-weight: bold;
        color: #3366CC;
        margin: 0;
        padding-left: 18px;
        text-decoration: none;
        text-align: left;
    }
    .botonesHover {
        text-decoration: none;
        color: #CC0000;
        background-color: #eeeeee;
    }
    /* Atención, evitar alterar la clase .subMe */
    .subMe{
        display: none;
        margin: 0;
        background-image: url(imasmenu/puntosvt.gif);
        background-repeat: repeat-y;
    }
    /* Atención, evitar alterar la clase .subMe */
    body {
        background-color: #FFB925;
        font-family: verdana, tahoma, arial, sans serif;
        font-size: 12px;
    }
-->
</style>
</head>
<body>
<table width="295" border="0" cellpadding="0" cellspacing="0">
  <tr>
    <td valign="top" bgcolor="#FFB925" id="tunMe">&nbsp;</td>
  </tr>
  <tr>
    <td width="295" valign="top" bgcolor="#FFB925"><p>&nbsp;</p></td>
  </tr>
</table>
</body>
</html>

```

### Conexión.php

```

<?php

//Establece la conexión con postgreSql
$connstr = "dbname=SICCDD user=postgres password=123
host=localhost port=5432";
$conexion = pg_connect($connstr);

?>

```

### Funcionaria.php

```

<html>
<head>
<title>Funcionaria - SISCON</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body>
<?php
include 'Conexion.php';
?>
<TABLE width=592 height=284

```

```

border=0 align=center cellPadding=0 cellSpacing=0>
<TBODY>
<TR>
<TD vAlign=top width=501><TABLE height=64 cellSpacing=0 cellPadding=0 width=576 align=center
border=0>
<TBODY>
<TR>
<TD width=576 height=64><form action="Ingreso_Funcionaria.php?valor=1" method="post"
name="frm_UsuarioFinal" id="frm_UsuarioFinal">
<p align="center"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
serif"><strong>FUNCIONARIA</strong></font></p>
<div align="right">
<table width="585" border="0" cellpadding="0" cellspacing="0">
<tr>
<td width="175"><div align="right"><font color="#002E5B" size="2" face="Verdana, Arial,
Helvetica, sans-serif">CEDULA:</font></div></td>
<td width="401">
<div align="left">
<input name="txt_CedulaF" type="text" id="txt_CedulaF" size="12" maxlength="10">
</div></td></tr>
<tr>
<td><p align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
serif">NOMBRES:</font></p></td>
<td><div align="left">
<input name="txt_NombresF" type="text" id="txt_NombresF" size="27" maxlength="25">
</div></td></tr>
<tr>
<td><div align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
serif">APELLIDOS:</font></div></td>
<td><div align="left">
<input name="txt_ApellidosF" type="text" id="txt_ApellidosF" size="27" maxlength="25">
</div></td>
</tr>
<tr>
<td><div align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
serif">DIRECCION:</font></div></td>
<td>
<div align="left">
<input name="txt_DireccionF" type="text" id="txt_DireccionF" size="57" maxlength="100">
</div></td>
</tr>
<tr>
<td><div align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
serif">TELEFONO:</font></div></td>
<td>
<div align="left">
<input name="txt_TelefonoF" type="text" id="txt_TelefonoF" size="32" maxlength="30">
</div></td>
</tr>
<tr>
<td><div align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica,
sans-serif">CELULAR:</font></div></td>
<td>
<div align="left">
<input name="txt_CelularF" type="text" id="txt_CelularF" size="32" maxlength="30">
</div></td>
</tr>
<tr>
<td><div align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
serif">USUARIO:</font></div></td>
<td>
<div align="left">
<input name="txt_UsuarioF" type="text" id="txt_UsuarioF" size="42" maxlength="40">
</div></td>
</tr>
<tr>

```



```

        <td height="28"><div align="right"><font color="#002E5B" size="2" face="Verdana, Arial,
Helvetica, sans-serif">PASSWORD:</font></div></td>
        <td>
            <div align="left">
                <input name="txt_PasswordF" type="text" id="txt_PasswordF" size="22" maxlength="20">
            </div></td>
        </tr>
        <tr>
            <td><div align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
serif">PRIVILEGIO:</font></div></td>
            <td>
                <div align="left">
                    <?php
                        $result = pg_exec("SELECT codigo_privilegio, descripcion_privilegio
FROM privilegio order by codigo_privilegio");
                        $filas = pg_num_rows($result);
                        echo "<select name='cmb_Privilegio'>";
                        for($i=0;$i<$filas;$i++){
                            $dato1 = pg_result($result,$i,0);
                            $dato2 = pg_result($result,$i,1);
                            echo "<option value='$dato1'>$dato2</option>";
                        }
                        echo "</select>";
                    ?>
                </div></td>
        </tr>
    </table>
</div>
<table width="260" border="0" align="center" cellpadding="0" cellspacing="0">
    <tr>
        <td width="260">&nbsp;</td>
        <td width="260">&nbsp;</td>
    </tr>
    <tr>
        <td><div align="center">
            <input name="btn_Grabar" type="submit" id="btn_Grabar" value="Guardar">
        </div></td>
        <td><div align="center">
            <input name="btn_Cancelar" type="reset" id="btn_Cancelar" value="Cancelar">
        </div></td>
    </tr>
</table>
<p align="center"><font size="2" face="Verdana, Arial, Helvetica, sans-serif"><a
href="Ingreso_Funcionaria.php?valor=3">Ver Funcionarias</a></font></p>
</form>
</TD>
</TR>
</TBODY>
</TABLE>
<DIV align=center></DIV></TD>
</TR>
</TBODY>
</TABLE>
</body>
</html>

```

#### Ingreso\_Funcionaria.php

```

<html>
<head>
<title>Funcionaria - SISCON</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body>
<TABLE height=176 cellSpacing=0 cellPadding=0 width=592 align=center border=0>
    <TBODY>

```

```

<TR>
  <TD width=91 height="176" vAlign=top
>&nbsp;   </TD>
  <TD vAlign=top width=501><TABLE height=64 cellSpacing=0 cellPadding=0 width=488 align=center
    border=0>
    <TBODY>
    <TR>
    <TD width=468 height=64><form action="Funcionaria.php" method="post" name="frm_Funcionaria2"
id="frm_Funcionaria2">
      <p align="center"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
serif"><strong>FUNCIONARIAS</strong></font></p>
      <div align="center">
        <table width="440" border="1" cellpadding="0" cellspacing="0">
          <tr bordercolor="#002E5B" bgcolor="#DCE0F1">
            <td width="105"><div align="center"><strong><font color="#002E5B" size="2" face="Verdana,
Arial, Helvetica, sans-serif">CEDULA</font></strong></div></td>
            <td width="329" bgcolor="#DCE0F1"><div align="center"><strong><font color="#002E5B"
size="2" face="Verdana, Arial, Helvetica, sans-serif">NOMBRES FUNCIONARIA</font></strong></div></td>
          </tr>
        </table>
      </div>
      <?php
include 'Conexion.php';
//Para Insertar Registros
$valor1=$HTTP_GET_VARS['valor'];
if($valor1==1){
  $CedulaF = $_POST[txt_CedulaF];
  $NombresF = $_POST[txt_NombresF];
  $ApellidosF = $_POST[txt_ApellidosF];
  $DireccionF = $_POST[txt_DireccionF];
  $TelefonoF = $_POST[txt_TelefonoF];
  $CelularF = $_POST[txt_CelularF];
  $UsuarioF = $_POST[txt_UsuarioF];
  $PasswordF = $_POST[txt_PasswordF];
  $Privilegio = $_POST[cmb_Privilegio];
  $v_codigofuncionaria = pg_exec("select nextval('seq_cod_func')");
  $filaSec = pg_numrows($v_codigofuncionaria);
  $v_codigofuncionaria1 = pg_result($v_codigofuncionaria,0,0);
  $Contador = 1;
  //Hace una consulta para ver si cliente existe
  $result=pg_exec("SELECT * FROM funcionaria where codigo_funcionaria = $v_codigofuncionaria1");
  //Ve el número de filas que obtiene el select si es 0 el cliente no existe
  $filas=pg_numrows($result);
  if($filas == 0){
    // Inserta los datos a la tabla
    $sql="INSERT INTO funcionaria (codigo_funcionaria, cedula_funcionaria, nombre_funcionaria,
apellido_funcionaria, direccion, telefono, celular, usuario, password, codigo_privilegio) VALUES
($v_codigofuncionaria1,$CedulaF,$NombresF,
$ApellidosF,$DireccionF,$TelefonoF,$CelularF,$UsuarioF,$PasswordF,$Privilegio)";
    pg_exec($conexion, $sql);
    // Muestra los datos de la tabla luego del ingreso del registro
    $result = pg_exec("SELECT cedula_funcionaria, nombre_funcionaria, apellido_funcionaria FROM funcionaria
order by nombre_funcionaria, apellido_funcionaria");
    $filas = pg_num_rows($result);
    for($i=0;$i<$filas;$i++){
      $dato1 = pg_result($result,$i,0);
      $dato2 = pg_result($result,$i,1);
      $dato3 = pg_result($result,$i,2);
      if(ceil($Contador/2) == ($Contador/2)){
        echo "<tr bgcolor='#FEF3FE' class='TextoTabla'>
          <td width='95'><div align='left'><a ref='Manten_Funcionaria.php?CedCli=$dato1'>$dato1
          </a></div></td>
          <td width='220'><div align='left'>$dato2 $dato3</div></td>
        </tr>";
      }else{
        echo "<tr bgcolor='#FFFFFF' class='TextoTabla'>

```

```

<td width='95' ><div align='left'><a href='Manten_Funcionaria.php?CedCli=$dato1'>$dato1
</a></div></td>
<td width='220'><div align='left'>$dato2 $dato3</div></td>
</tr>";
}
$Contador++;
}
}else{
echo "<p align='center' class='TextoTabla1'>Funcionaria ya existe: Datos no se grabaron</p>";
}
}

//Para Modificar registros
if($valor1==2){
$CedulaF = $_POST[txt_CedulaF];
$NombresF = $_POST[txt_NombresF];
$ApellidosF = $_POST[txt_ApellidosF];
$DireccionF = $_POST[txt_DireccionF];
$TelefonoF = $_POST[txt_TelefonoF];
$CelularF = $_POST[txt_CelularF];
$UsuarioF = $_POST[txt_UsuarioF];
$PasswordF = $_POST[txt_PasswordF];
$Privilegio = $_POST[cmb_Privilegio];
$Contador = 1;
// Modifica los datos a la tabla
$sql = "UPDATE funcionaria SET nombre_funcionaria = '$NombresF', apellido_funcionaria =
'$ApellidosF', direccion = '$DireccionF', telefono = '$TelefonoF', celular = '$CelularF' WHERE
cedula_funcionaria = '$CedulaF'";
pg_exec($conexion, $sql);
// Muestra los datos de la tabla luego del ingreso del registro
$result = pg_exec("SELECT cedula_funcionaria, nombre_funcionaria, apellido_funcionaria
FROM funcionaria order by nombre_funcionaria, apellido_funcionaria");
$filas = pg_num_rows($result);
for($i=0;$i<$filas;$i++){
$dato1 = pg_result($result,$i,0);
$dato2 = pg_result($result,$i,1);
$dato3 = pg_result($result,$i,2);
if(ceil($Contador/2) == ($Contador/2)){
echo "<tr bgcolor='#FEF3FE' class='TextoTabla'>
td width='95' ><div align='left'><a href='Manten_Funcionaria.php?CedCli=$dato1'>$dato1
</a></div></td>
<td width='220'><div align='left'>$dato2 $dato3</div></td>
</tr>";
}
}else{
echo "<tr bgcolor='#FFFFFF' class='TextoTabla'>
<td width='95' ><div align='left'><a href='Manten_Funcionaria.php?CedCli=$dato1'>$dato1
</a></div></td>
<td width='220'><div align='left'>$dato2 $dato3</div></td>
</tr>";
}
$Contador++;
}
}

//Para ver registros
if($valor1==3){
//$CedulaF = $_POST[txt_CedulaF];
$NombresF = $_POST[txt_NombresF];
$DireccionF = $_POST[txt_DireccionF];
$TelefonoF = $_POST[txt_TelefonoF];
$CelularF = $_POST[txt_CelularF];
$UsuarioF = $_POST[txt_UsuarioF];
$PasswordF = $_POST[txt_PasswordF];
$Contador = 1;
// Muestra los datos de la tabla luego del ingreso del registro

```



```

?>
<TABLE width=592 height=259
  border=0 align=center cellPadding=0 cellSpacing=0>
  <TBODY>
    <TR>
      <TD width=501 height="259" vAlign=top><TABLE height=64 cellSpacing=0 cellPadding=0 width=725
        align=center
        border=0>
        <TBODY>
          <TR>
            <TD width=725 height=64><form action="Ingreso_Funcionaria.php?valor=2" method="post"
              name="frm_UsuarioFinal2" id="frm_UsuarioFinal2">
              <p align="center"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
                serif"><strong>FUNCIONARIAS</strong></font></p>
              <div align="center">
                <table width="636" border="0" cellpadding="0" cellspacing="0">
                  <tr>
                    <td width="160"><div align="right"><font color="#002E5B" size="2" face="Verdana, Arial,
                      Helvetica, sans-serif">CEDULA:</font></div></td>
                    <td width="476"><div align="left"><font color="#002E5B" size="2" face="Verdana, Arial,
                      Helvetica, sans-serif"><?php echo "$CedulaF1"; ?>
                      <?php $CedulaF1= pg_result($result, 0, 1);
                        echo "<input name='txt_CedulaF' type='hidden' id='txt_CedulaF'
                          value='$CedulaF1'>";
                    </td>
                  </tr>
                </table>
              </div>
            </td>
          </tr>
          <tr>
            <td><p align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
              serif">NOMBRES:</font></p></td>
            <td><div align="left">
              <?php $DireccionF1 = pg_result($result, 0, 2);
              echo "<input name='txt_NombresF' type='text' id='txt_NombresF' size='27' maxlength='25'
                value='$DireccionF1'>";
            </div>
            </td>
          </tr>
          <tr>
            <td><div align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
              serif">APELLIDOS:</font></div></td>
            <td><div align="left"><?php $ApellidosF1 = pg_result($result, 0, 3);
              echo "<input name='txt_ApellidosF' type='text' id='txt_ApellidosF' size='27' maxlength='25'
                value='$ApellidosF1'>";
            </div>
            </td>
          </tr>
          <tr>
            <td><div align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
              serif">DIRECCION:</font></div></td>
            <td><div align="left">
              <?php $TelefonoF1 = pg_result($result, 0, 4);
              echo "<input name='txt_DireccionF' type='text' id='txt_DireccionF' size='57'
                maxlength='100' value='$TelefonoF1'>";
            </div>
            </td>
          </tr>
          <tr>
            <td><div align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
              serif">TELEFONO:</font></div></td>
            <td><div align="left">
              <?php $TelefonoF1 = pg_result($result, 0, 5);
              echo "<input name='txt_TelefonoF' type='text' id='txt_TelefonoF' size='32' maxlength='30'
                value='$TelefonoF1'>";
            </div>
            </td>
          </tr>
        </tr>
      </td>
    </tr>
  </tbody>
</table>

```

```

        <td><div align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
        serif">CELULAR:</font></div></td>
        <td> <div align="left"> <?php $CelularF1 = pg_result($result, 0, 6);
        echo "<input name='txt_CelularF' type='text' id='txt_CelularF' size='32' maxlength='30'
        value='$CelularF1'>";
        ?>
    </div></td> </tr>
</table>
<?php
        pg_FreeResult($result);
    ?>
    <table width="260" border="0" cellpadding="0" cellspacing="0">
        <tr>
            <td width="260">&nbsp;</td>
        </tr>
        <tr>
            <td><div align="center">
                <input name="btn_Grabar" type="submit" id="btn_Grabar" value="Guardar Cambios">
            </div></td>
        </tr>
    </table>
    <p><font size="2" face="Verdana, Arial, Helvetica, sans-serif"><a
    href="Ingreso_Funcionaria.php?valor=3">Ver Funcionarias</a></font></p>
</div>
</form>
</TD>
</TR>
</TBODY>
</TABLE>
<DIV align=center></DIV></TD>
</TR>
</TBODY>
</TABLE>
</body>
</html>

```

#### Ingreso\_ObjetoTransaccion.php

```

<html>
<head>
<title>Tipos de Transaccion - SISCON</title>
</head>
<body>
<TABLE height=64 cellSpacing=0 cellPadding=0 width=474 align=center
border=0>
<TBODY>
<TR>
<TD width=474 height=64><form action="ObjetoTransaccion.html" method="post"
name="frm_Funcionaria2" id="frm_Funcionaria2">
<p align="center"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
serif"><strong>TIPOS DE TRANSACCION </strong></font></p>
<div align="center">
<table width="440" border="1" cellpadding="0" cellspacing="0">
<tr bordercolor="#002E5B" bgcolor="#DCE0F1">
<td width="105"><div align="center"><strong><font color="#002E5B" size="2" face="Verdana, Arial,
Helvetica, sans-serif">CODIGO</font></strong></div></td>
<td width="329" bgcolor="#DCE0F1"><div align="center"><strong><font color="#002E5B" size="2"
face="Verdana, Arial, Helvetica, sans-serif">DESCRIPCION</font></strong></div></td>
</tr>
</table>
<?php
        include 'Conexion.php';
        //Para Insertar Registros
        $valor1=$_HTTP_GET_VARS['valor'];
        if($valor1==1)
        {

```

```

$CodigoTT = $_POST[txt_CodigoTT];
$DescripcionTT = $_POST[txt_DescripcionTT];
$v_codigott = pg_exec("select nextval('seq_cod_ttra')");
$filaSec = pg_numrows($v_codigott);
$v_codigott1 = pg_result($v_codigott,0,0);
$Contador = 1;
//Hace una consulta para ver si cliente existe
$result=pg_exec("SELECT * FROM objetotransaccion where codigo_ttransaccion =
$v_codigott1");
//Ve el número de filas que obtiene el select si es 0 el cliente no existe
$filas=pg_numrows($result);
if($filas == 0){
// Inserta los datos a la tabla
$sql ="INSERT INTO objetotransaccion (codigo_ttransaccion, descripcion_ttransaccion)
VALUES ($v_codigott1,$DescripcionTT)";
pg_exec($conexion, $sql);
// Muestra los datos de la tabla luego del ingreso del registro
$result = pg_exec("SELECT codigo_ttransaccion, descripcion_ttransaccion FROM
objetotransaccion order by codigo_ttransaccion");
$filas = pg_num_rows($result);
for($i=0;$i<$filas;$i++){
$dato1 = pg_result($result,$i,0);
$dato2 = pg_result($result,$i,1);
if(ceil($Contador/2) == ($Contador/2)){
echo "<tr bgcolor=#FEF3FE' class='TextoTabla'>
<td width='95' ><div align='left'><a
href='Manten_ObjetoTransaccion.php?CedCli=$dato1'>$dato1</a></div></td>
<td width='220'><div align='left'>$dato2</div></td>
</tr>";
}else{
echo "<tr bgcolor=#FFFFFF' class='TextoTabla'>
<td width='95' ><div align='left'><a href= 'Manten_ObjetoTransaccion.php?CedCli=$dato1'
> $dato1</a></div></td>
<td width='220'><div align='left'>$dato2</div></td>
</tr>";
}
}
$Contador++;
}
}else{
echo "<p align='center' class='TextoTabla1'>Tipo Transaccion ya existe: Datos no se
grabaron</p>";
}
}
//Para Modificar registros
if($valor1==2)
{
$CodigoTT = $_POST[txt_CodigoTT];
$DescripcionTT = $_POST[txt_DescripcionTT];
$Contador = 1;
// Modifica los datos a la tabla
$sql ="UPDATE objetotransaccion SET descripcion_ttransaccion = '$DescripcionTT'
WHERE codigo_ttransaccion = $CodigoTT";
pg_exec($conexion, $sql);
// Muestra los datos de la tabla luego del ingreso del registro
$result = pg_exec("SELECT codigo_ttransaccion, descripcion_ttransaccion FROM
objetotransaccion order by codigo_ttransaccion");
$filas = pg_num_rows($result);
for($i=0;$i<$filas;$i++){
$dato1 = pg_result($result,$i,0);
$dato2 = pg_result($result,$i,1);
if(ceil($Contador/2) == ($Contador/2)){
echo "<tr bgcolor=#FEF3FE' class='TextoTabla'>
<td width='95' ><div align='left'><a
href='Manten_ObjetoTransaccion.php?CedCli=$dato1'>$dato1</a></div></td>

```





```

<html>
<head>
<title>Tipos de Documento - SISCON</title>
</head>
<body>
<?php
    $CodigoTT1 = $_HTTP_GET_VARS['CedCli'];
    $Contador = 1;
    include 'Conexion.php';
    //Hace una consulta para ver datos de la habitacion
    $result=pg_exec("SELECT * FROM objetotransaccion where codigo_ttransaccion = $CodigoTT1");
    $filas=pg_numrows($result);
?>
<TABLE width=592 height=259
    border=0 align=center cellPadding=0 cellSpacing=0>
    <TBODY>
    <TR>
    <TD width=501 height="259" vAlign=top><TABLE height=64 cellSpacing=0 cellPadding=0 width=725
align=center
    border=0>
    <TBODY>
    <TR>
    <TD width=725 height=64><form action="Ingreso_ObjetoTransaccion.php?valor=2" method="post"
name="frm_UsuarioFinal2" id="frm_UsuarioFinal2">
    <p align="center"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
serif"><strong>TIPOS DE TRANSACCION </strong></font></p>
    <div align="center">
    <table width="636" border="0" cellpadding="0" cellspacing="0">
    <tr>
    <td width="160"><div align="left"><font color="#002E5B" size="2" face="Verdana, Arial,
Helvetica, sans-serif">CODIGO:</font></div></td>
    <td width="476"><div align="left"><font color="#002E5B" size="2" face="Verdana, Arial,
Helvetica, sans-serif"><?php echo "$CodigoTT1"; ?>
    <?php $CodigoTT1= pg_result($result, 0, 0);
        echo "<input name='txt_CodigoTT' type='hidden' id='txt_CodigoTT'
        value='$CodigoTT1'>";
    ?>
    </font></div></td>
    </tr>
    <tr>
    <td><p align="left"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
serif">DESCRIPCION:</font></p></td>
    <td><div align="left">
    <?php $DescripcionTT1 = pg_result($result, 0, 1);
        echo "<input name='txt_DescripcionTT' type='text' id='txt_DescripcionTT'
        size='55' maxlength='50' value='$DescripcionTT1'>";
    ?>
    </div></td>
    </tr>
    </tr>
    </table>
    <?php
        pg_FreeResult($result);
?>
    <table width="260" border="0" cellpadding="0" cellspacing="0">
    <tr>
    <td width="260">&nbsp;</td>
    </tr>
    <tr>
    <td><div align="center">
    <input name="btn_Grabar" type="submit" id="btn_Grabar" value="Guardar Cambios">
    </div></td>
    </tr>
    </table>

```

```

        <p><font size="2" face="Verdana, Arial, Helvetica, sans-serif"><a
href="Ingreso_ObjetoTransaccion.php?valor=3">Ver Tipos de Transaccion </a></font></p>
    </div>
</form>
</TD>
</TR>
</TBODY>
</TABLE>
<DIV align=center></DIV></TD>
</TR>
</TBODY>
</TABLE>
</body>
</html>

```

### Ingreso\_UsuarioFinal.php

```

<html>
<head>
<title>Usuario Final - SISCON</title>
</head>
<body>
<TABLE height=284 cellSpacing=0 cellPadding=0 width=592 align=center
border=0>
<TBODY>
<TR>
<TD vAlign=top width=91 >&nbsp;</TD>
<TD vAlign=top width=501><TABLE height=64 cellSpacing=0 cellPadding=0 width=488 align=center
border=0>
<TBODY>
<TR>
<TD width=468 height=64><form action="UsuarioFinal.html" method="post" name="frm_Cliente1"
id="frm_Cliente1">
<p align="center"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
serif"><strong>USUARIO FINAL</strong></font></p>
<div align="center">
<table width="440" border="1" cellpadding="0" cellspacing="0">
<tr bordercolor="#002E5B" bgcolor="#DCE0F1">
<td width="105"><div align="center"><strong><font color="#002E5B" size="2" face="Verdana,
Arial, Helvetica, sans-serif">CODIGO</font></strong></div></td>
<td width="329" bgcolor="#DCE0F1"><div align="center"><strong><font color="#002E5B"
size="2" face="Verdana, Arial, Helvetica, sans-serif">DESCRIPCION</font></strong></div></td>
</tr>
</table>
<?php
include 'Conexion.php';
//Para Insertar Registros
$valor1=$_HTTP_GET_VARS['valor'];
if($valor1==1)
{
$DescripcionUF = $_POST[txt_DescripcionUF];
$v_codigouf = pg_exec("select nextval('seq_cod_uf')");
$filaSec = pg_numrows($v_codigouf);
$v_codigouf1 = pg_result($v_codigouf,0,0);
$Contador = 1;
//Hace una consulta para ver si cliente existe
$result=pg_exec("SELECT * FROM usuario_final where codigo_ufinal = $v_codigouf1");
//Ve el n&uacute;mero de filas que obtiene el select si es 0 el cliente no existe
$filas=pg_numrows($result);
if ($filas==0)
{
// Inserta los datos a la tabla
$sql ="INSERT INTO usuario_final (codigo_ufinal, descripcion_ufinal) VALUES
($v_codigouf1,$DescripcionUF)";
pg_exec($conexion, $sql);
// Muestra los datos de la tabla luego del ingreso del registro

```

```

$result = pg_exec("SELECT codigo_ufinal, descripcion_ufinal FROM usuario_final order
by codigo_ufinal");
$filas = pg_num_rows($result);
for($i=0;$i<$filas;$i++){
    $dato1 = pg_result($result,$i,0);
    $dato2 = pg_result($result,$i,1);
    if(ceil($Contador/2) == ($Contador/2))
    {
        echo "<tr bgcolor='#FEF3FE' class='TextoTabla'>
        <td width='95' ><div align='left'><a
        href='Manten_UsuarioFinal.php?CedCli=$dato1'>$dato1 </a></div></td>
        <td width='220'><div align='left'>$dato2</div></td>
        </tr>";
    }
    else
    {
        echo "<tr bgcolor='#FFFFFF' class='TextoTabla'>
        <td width='95' ><div align='left'><a
        href='Manten_UsuarioFinal.php?CedCli=$dato1'>$dato1 </a></div></td>
        <td width='220'><div align='left'>$dato2</div></td>
        </tr>";
    }
    $Contador++;
}
}
else
{
    echo "<p align='center' class='TextoTabla1'>Usuario Final ya existe: Datos no se
    grabaron</p>";
}
}
//Para Modificar registros
if($valor1==2)
{
    $CodigoUF = $_POST[txt_CodigoUF];
    $DescripcionUF = $_POST[txt_DescripcionUF];
    $Contador = 1;
    // Modifica los datos a la tabla
    $sql ="UPDATE usuario_final SET descripcion_ufinal = '$DescripcionUF' WHERE
    codigo_ufinal = $CodigoUF";
    pg_exec($conexion, $sql);
    // Muestra los datos de la tabla luego del ingreso del registro
    $result = pg_exec("SELECT codigo_ufinal, descripcion_ufinal FROM usuario_final order
    by codigo_ufinal");
    $filas = pg_num_rows($result);
    for($i=0;$i<$filas;$i++){
        $dato1 = pg_result($result,$i,0);
        $dato2 = pg_result($result,$i,1);
        if(ceil($Contador/2) == ($Contador/2))
        {
            echo "<tr bgcolor='#FEF3FE' class='TextoTabla'>
            <td width='95' ><div align='left'><a
            href='Manten_UsuarioFinal.php?CedCli=$dato1'>$dato1 </a></div></td>
            <td width='220'><div align='left'>$dato2</div></td>
            </tr>";
        }
        else
        {
            echo "<tr bgcolor='#FFFFFF' class='TextoTabla'>
            <td width='95' ><div align='left'><a
            href='Manten_UsuarioFinal.php?CedCli=$dato1'>$dato1 </a></div></td>
            <td width='220'><div align='left'>$dato2</div></td>
            </tr>";
        }
    }
    $Contador++;
}

```



```

<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body>
<?php
    $CodigoUF1 = $_HTTP_GET_VARS['CedCli'];
    $Contador = 1;
    //Establece la conexión con mysql
    include 'Conexion.php';
    //Hace una consulta para ver datos de la habitacion
    $result=pg_exec("SELECT * FROM usuario_final where codigo_ufinal = $CodigoUF1");
    $filas=pg_numrows($result);
?>
<TABLE width=592 height=284
    border=0 align=center cellPadding=0 cellSpacing=0>
<TBODY>
<TR>
<TD vAlign=top width=501><TABLE height=64 cellSpacing=0 cellPadding=0 width=488 align=center
    border=0>
<TBODY>
<TR>
<TD width=468 height=64><form action="Ingreso_UsuarioFinal.php?valor=2" method="post"
    name="frm_UsuarioFinal2" id="frm_UsuarioFinal2">
<p align="center"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
    serif"><strong>USUARIO FINAL </strong></font></p>
<div align="center">
<table width="424" border="0" cellpadding="0" cellspacing="0">
<tr>
<td width="112"><div align="left"><font color="#002E5B" size="2" face="Verdana, Arial,
    Helvetica, sans-serif">CODIGO:</font></div></td>
<td width="312"><div align="left"><font color="#002E5B" size="2" face="Verdana, Arial,
    Helvetica, sans-serif"><?php echo "$CodigoUF1"; ?>
<?php $CodigoUF1= pg_result($result, 0, 0);
    echo "<input name='txt_CodigoUF' type='hidden' id='txt_CodigoUF'
    value='$CodigoUF1'>";
    ?>
</font></div></td>
</tr>
<tr>
<td><p align="left"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
    serif">DESCRIPCION:</font></p></td>
<td><div align="left">
<?php $DescripcionUF1 = pg_result($result, 0, 1);
    echo "<input name='txt_DescripcionUF' type='text' id='txt_DescripcionUF' size='65'
    maxlength='100' value='$DescripcionUF1'>";
    ?>
</div></td>
</tr>
</table>
<?php
    pg_FreeResult($result);
?>
<table width="260" border="0" cellpadding="0" cellspacing="0">
<tr>
<td width="260">&nbsp;</td>
</tr>
<tr>
<td><div align="center">
<input name="btn_Grabar" type="submit" id="btn_Grabar" value="Guardar Cambios">
</div></td>
</tr>
</table>
<p><font size="2" face="Verdana, Arial, Helvetica, sans-serif"><a
href="Ingreso_UsuarioFinal.php?valor=3">Ver Usuarios Finales </a></font></p>
</div>

```

```

        <p align="center">&nbsp;</p>
    </form>
    <div align="justify"></div></TD>
</TR>
</TBODY>
</TABLE>
<DIV align=center></DIV></TD>
</TR>
</TBODY>
</TABLE>
</body>
</html>

```

### Ingreso\_TipoDocumento.php

```

<html>
<head>
<title>Tipos de Documento - SISCON</title>
</head>
<body>
<TABLE height=64 cellSpacing=0 cellPadding=0 width=474 align=center
border=0>
<TBODY>
<TR>
<TD width=474 height=64><form action="TipoDocumento.html" method="post" name="frm_Funcionaria2"
id="frm_Funcionaria2">
<p align="center"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
serif"><strong>TIPOS DE DOCUMENTO</strong></font></p>
<div align="center">
<table width="440" border="1" cellpadding="0" cellspacing="0">
<tr bordercolor="#002E5B" bgcolor="#DCE0F1">
<td width="105"><div align="center"><strong><font color="#002E5B" size="2" face="Verdana, Arial,
Helvetica, sans-serif">CODIGO</font></strong></div></td>
<td width="329" bgcolor="#DCE0F1"><div align="center"><strong><font color="#002E5B" size="2"
face="Verdana, Arial, Helvetica, sans-serif">DESCRIPCION</font></strong></div></td>
</tr>
</table>
<?php
include 'Conexion.php';
//Para Insertar Registros
$valor1=$_HTTP_GET_VARS['valor'];
if($valor1==1)
{
    $CodigoTD = $_POST[txt_CodigoTD];
    $DescripcionTD = $_POST[txt_DescripcionTD];
    $v_codigotd = pg_exec("select nextval('seq_cod_tdoc')");
    $filaSec = pg_numrows($v_codigotd);
    $v_codigotd1 = pg_result($v_codigotd,0,0);
    $Contador = 1;
    //Hace una consulta para ver si cliente existe
    $result=pg_exec("SELECT * FROM tipodocumento where codigo_documento =
    $v_codigotd1");
    //Ve el número de filas que obtiene el select si es 0 el cliente no existe
    $filas=pg_numrows($result);
    if($filas == 0){
        // Inserta los datos a la tabla
        $sql ="INSERT INTO tipodocumento (codigo_documento, descripcion_documento)
        VALUES ($v_codigotd1,$DescripcionTD)";

        pg_exec($conexion, $sql);
        // Muestra los datos de la tabla luego del ingreso del registro
        $result = pg_exec("SELECT codigo_documento, descripcion_documento FROM tipodocumento
        order by codigo_documento");
        $filas = pg_num_rows($result);
        for($i=0;$i<$filas;$i++){
            $dato1 = pg_result($result,$i,0);

```

```

$dato2 = pg_result($result,$i,1);
if(ceil($Contador/2) == ($Contador/2)){
echo "<tr bgcolor='#FEF3FE' class='TextoTabla'>
<td width='95' ><div align='left'><a
href='Manten_TipoDocumento.php?CedCli=$dato1'>$dato1</a></div></td>
<td width='220'><div align='left'>$dato2</div></td>
</tr>";
}else{
echo "<tr bgcolor='#FFFFFF' class='TextoTabla'>
<td width='95' ><div align='left'><a
href='Manten_TipoDocumento.php?CedCli=$dato1'>$dato1</a></div></td>
<td width='220'><div align='left'>$dato2</div></td>
</tr>";
}
$Contador++;
}
}else{
echo "<p align='center' class='TextoTabla1'>Tipo Documento ya existe: Datos no se
grabaron</p>";
}
}
//Para Modificar registros
if($valor1==2)
{
$CodigoTD = $_POST[txt_CodigoTD];
$DescripcionTD = $_POST[txt_DescripcionTD];
$Contador = 1;
// Modifica los datos a la tabla
$sql="UPDATE tipodocumento SET descripcion_tdocumento = '$DescripcionTD' WHERE
codigo_tdocumento = '$CodigoTD'";
pg_exec($conexion, $sql);
// Muestra los datos de la tabla luego del ingreso del registro
$result = pg_exec("SELECT codigo_tdocumento, descripcion_tdocumento FROM tipodocumento
order by codigo_tdocumento");
$filas = pg_num_rows($result);
for($i=0;$i<$filas;$i++){
$dato1 = pg_result($result,$i,0);
$dato2 = pg_result($result,$i,1);
if(ceil($Contador/2) == ($Contador/2)){
echo "<tr bgcolor='#FEF3FE' class='TextoTabla'>
<td width='95' ><div align='left'><a
href='Manten_TipoDocumento.php?CedCli=$dato1'>$dato1</a></div></td>
<td width='220'><div align='left'>$dato2</div></td>
</tr>";
}else{
echo "<tr bgcolor='#FFFFFF' class='TextoTabla'>
<td width='95' ><div align='left'><a
href='Manten_TipoDocumento.php?CedCli=$dato1'>$dato1</a></div></td>
<td width='220'><div align='left'>$dato2</div></td>
</tr>";
}
$Contador++;
}}
//Para ver registros
if($valor1==3)
{
$CodigoTD = $_POST[txt_CodigoTD];
$DescripcionTD = $_POST[txt_DescripcionTD];
$Contador = 1;
Muestra los datos de la tabla luego del ingreso del registro
$result = pg_exec("SELECT codigo_tdocumento, descripcion_tdocumento FROM tipodocumento
order by codigo_tdocumento");
$filas = pg_num_rows($result);

```

```

for($i=0;$i<$filas;$i++){
    $dato1 = pg_result($result,$i,0);
    $dato2 = pg_result($result,$i,1);
    if(ceil($Contador/2) == ($Contador/2)){
        echo "<tr bgcolor=#FEF3FE class='TextoTabla'>
        <td width='95' ><div align='left'><a
        href='Manten_TipoDocumento.php?CedCli=$dato1'>$dato1 </a></div></td>
        <td width='220'><div align='left'>$dato2</div></td>
        </tr>";
    }else{
        echo "<tr bgcolor='#FFFFFF' class='TextoTabla'>
        <td width='95' ><div align='left'><a
        href='Manten_TipoDocumento.php?CedCli=$dato1'>$dato1 </a></div></td>
        <td width='220'><div align='left'>$dato2</div></td>
        </tr>";
    }
    $Contador++;
}
pg_FreeResult($result);
</table>
<table width="200" border="0" cellpadding="0" cellspacing="0">
<tr>
<td>&nbsp;</td>
</tr>
<tr>
<td><div align="center">
<input name="btn_Volver" type="submit" id="btn_Volver" value="Volver">
</div>
<div align="center"></div></td>
</tr>
</table>
<p>&nbsp;</p>
</div>
</form>
</TD>
</TR>
</TBODY>
</TABLE>
</body>
</html>

```

#### **Manten\_TipoDocumento.php**

```

<html>
<head>
<title>Tipos de Documento - SISCON</title>
</head>
<body>
<?php
    $CodigoTD1 = $_HTTP_GET_VARS['CedCli'];
    $Contador = 1;
    //Establece la conexión con mysql
    include 'Conexion.php';
    //Hace una consulta para ver datos de la habitacion
    $result=pg_exec("SELECT * FROM tipodocumento where codigo_tdocumento = '$CodigoTD1'");
    $filas=pg_numrows($result);
?>
<TABLE width=592 height=259
    border=0 align=center cellPadding=0 cellSpacing=0>
<TBODY>
<TR>
<TD width=501 height="259" vAlign=top><TABLE height=64 cellSpacing=0 cellPadding=0 width=725
    align=center
    border=0>
<TBODY>

```



```

<TR>
<TD width=725 height=64><form action="Ingreso_TipoDocumento.php?valor=2" method="post"
name="frm_UsuarioFinal2" id="frm_UsuarioFinal2">
<p align="center"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
serif"><strong>TIPOS DE DOCUMENTO</strong></font></p>
<div align="center">
<table width="636" border="0" cellpadding="0" cellspacing="0">
<tr>
<td width="160"><div align="left"><font color="#002E5B" size="2" face="Verdana, Arial,
Helvetica, sans-serif">CODIGO:</font></div></td>
<td width="476"><div align="left"><font color="#002E5B" size="2" face="Verdana, Arial,
Helvetica, sans-serif"><?php echo "$CodigoTD1"; ?>
<?php $CodigoTD1= pg_result($result, 0, 0);
echo "<input name='txt_CodigoTD' type='hidden' id='txt_CodigoTD'
value='$CodigoTD1'>";
?>
</font></div></td>
</tr>
<tr>
<td><p align="left"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
serif">DESCRIPCION:</font></p></td>
<td><div align="left">
<?php $DescripcionTD1 = pg_result($result, 0, 1);
echo "<input name='txt_DescripcionTD' type='text' id='txt_DescripcionTD' size='55'
maxlength='50' value='$DescripcionTD1'>";
?>
</div></td>
</tr>
</table>
<?php
pg_FreeResult($result);
?>
<table width="260" border="0" cellpadding="0" cellspacing="0">
<tr>
<td width="260">&nbsp;</td>
</tr>
<tr>
<td><div align="center">
<input name="btn_Grabar" type="submit" id="btn_Grabar" value="Guardar Cambios">
</div></td>
</tr>
</table>
<p><font size="2" face="Verdana, Arial, Helvetica, sans-serif"><a
href="Ingreso_TipoDocumento.php?valor=3">Ver Tipos de Documento </a></font></p>
</div>
</form>
</TD>
</TR>
</TBODY>
</TABLE>
<DIV align=center></DIV></TD>
</TR>
</TBODY>
</TABLE>
</body>
</html>

```

#### Productos.php

```

<html>
<head>
<title>Productos - SISCON</title>
</head>
<body>
<?php

```

```

include 'Conexion.php';
?>
<TABLE width=592 height=284
  border=0 align=center cellPadding=0 cellSpacing=0>
  <TBODY>
    <TR>
      <TD vAlign=top width=501><TABLE height=64 cellSpacing=0 cellPadding=0 width=576 align=center
        border=0>
          <TBODY>
            <TR>
              <TD width=576 height=64><form action="Ingreso_Productos.php?valor=1" method="post"
name="frm_UsuarioFinal" id="frm_UsuarioFinal">
                <p align="center"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
serif"><strong>PRODUCTOS</strong></font></p>
                <div align="right">
                  <table width="585" border="0" cellpadding="0" cellspacing="0">
                    <tr>
                      <td width="175"><div align="right"><font color="#002E5B" size="2" face="Verdana, Arial,
Helvetica, sans-serif">CODIGO:</font></div></td>
                      <td width="401">
                        <div align="left">
                          <input name="txt_CodigoProd" type="text" id="txt_CodigoProd" size="10" maxlength="5">
                        </div></td></tr>
                      <tr>
                        <td><p align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
serif">DESCRIPCION:</font></p></td>
                        <td>
                          <div align="left">
                            <input name="txt_DescripcionProd" type="text" id="txt_DescripcionProd" size="60"
maxlength="100">
                          </div></td></tr>
                      <tr>
                        <td><div align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
serif">STOCK MAXIMO:</font></div></td>
                        <td>
                          <div align="left">
                            <input name="txt_StockMax" type="text" id="txt_StockMax" size="20" maxlength="15">
                          </div></td>
                        </tr>
                      <tr>
                        <td><div align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
serif">STOCK MINIMO :</font></div></td>
                        <td>
                          <div align="left">
                            <input name="txt_StockMin" type="text" id="txt_StockMin" size="20" maxlength="15">
                          </div></td>
                        </tr>
                      <tr>
                        <td><div align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
serif">EXISTENCIA:</font></div></td>
                        <td>
                          <div align="left">
                            <input name="txt_Existencia" type="text" id="txt_Existencia" size="20" maxlength="15">
                          </div></td>
                        </tr>
                      <tr>
                        <td><div align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
serif">TIPO DE UNIDAD:</font></div></td>
                        <td>
                          <div align="left">
                            <?php
$result = pg_exec("SELECT codigo_tunidad, descripcion_tunidad
FROM tipounidad
order by codigo_tunidad");
filas = pg_num_rows($result);
//echo "<h1> Diego $filas <h1>";

```

```

        echo "<select name='cmb_TUnidad'>";
        for($i=0;$i<$filas;$i++)
        {
            $dato1 = pg_result($result,$i,0);
            $dato2 = pg_result($result,$i,1);
            echo "<option value='$dato1'>$dato2</option>";
        }
        echo "</select>";
    ?>
</div></td>          </tr>
<tr>
<td><div align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">TIPO DE PRODUCTO:</font></div></td>
<td><div align="left">
    <?php
        $result1 = pg_exec("SELECT codigo_tproducto, descripcion_tproducto
        FROM tipoproducto
        order by codigo_tproducto");
        $filas1 = pg_num_rows($result1);
        echo "<select name='cmb_TProducto'>";
        for($i=0;$i<$filas;$i++)
        {
            $dato1_1 = pg_result($result1,$i,0);
            $dato2_1 = pg_result($result1,$i,1);
            echo "<option value='$dato1_1'>$dato2_1</option>";
        }
        echo "</select>";
    ?>
    </div></td>
</tr>
</table>
</div>
<table width="260" border="0" align="center" cellpadding="0" cellspacing="0">
<tr>
<td width="260">&nbsp;</td>
<td width="260">&nbsp;</td>
</tr>
<tr>
<td><div align="center">
    <input name="btn_Grabar" type="submit" id="btn_Grabar" value="Guardar">
</div></td>
<td><div align="center">
    <input name="btn_Cancelar" type="reset" id="btn_Cancelar" value="Cancelar">
</div></td>
</tr>
</table>
<p align="center"><font size="2" face="Verdana, Arial, Helvetica, sans-serif"><a
href="Ingreso_Productos.php?valor=3">Ver Productos</a></font></p>
</form>
</TD>
</TR>
</TBODY>
</TABLE>
<DIV align=center></DIV></TD>
</TR>
</TBODY>
</TABLE>
</body>
</html>

Ingreso_Productos.php
<html>
<head>
<title>Productos - SISCON</title>

```

```

</head>
<body>
<TABLE height=176 cellSpacing=0 cellPadding=0 width=592 align=center
border=0>
<TBODY>
<TR>
<TD width=91 height="176" vAlign=top;</TD>
<TD vAlign=top width=501><TABLE height=64 cellSpacing=0 cellPadding=0 width=488 align=center
border=0>
<TBODY>
<TR>
<TD width=468 height=64><form action="Productos.php" method="post" name="frm_Funcionaria2"
id="frm_Funcionaria2">
<p align="center"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
serif"><strong>PRODUCTOS</strong></font></p>
<div align="center">
<table width="440" border="1" cellpadding="0" cellspacing="0">
<tr bordercolor="#002E5B" bgcolor="#DCE0F1">
<td width="105"><div align="center"><strong><font color="#002E5B" size="2" face="Verdana,
Arial, Helvetica, sans-serif">CODIGO</font></strong></div></td>
<td width="329" bgcolor="#DCE0F1"><div align="center"><strong><font color="#002E5B"
size="2" face="Verdana, Arial, Helvetica, sans-
serif">DESCRIPCION</font></strong></div></td>
</tr>
<?php
'Conexion.php';
//Para Insertar Registros
$valor1=$_HTTP_GET_VARS['valor'];
if($valor1==1)
{
$CodigoProd = $_POST[txt_CodigoProd];
$DescripcionProd = $_POST[txt_DescripcionProd];
$StockMax = $_POST[txt_StockMax];
$StockMin = $_POST[txt_StockMin];
$Existencia = $_POST[txt_Existencia];
$TUnidad = $_POST[cmb_TUnidad];
$TProducto = $_POST[cmb_TProducto];
$Contador = 1;
//Hace una consulta para ver si cliente existe
$result=pg_exec("SELECT * FROM producto where codigo_producto = 'CodigoProd'");
//Ve el n&uacute;mero de filas que obtiene el select si es 0 el cliente no existe
filas=pg_numrows($result);
if($filas == 0){
// Inserta los datos a la tabla
$sql="INSERT INTO producto (codigo_producto, descripcion_producto, stock_maximo,
stock_minimo, existencia, codigo_tunidad, codigo_tproducto) VALUES ('$CodigoProd',
'$DescripcionProd', '$StockMax', '$StockMin', '$Existencia', '$TUnidad', '$TProducto')";
pg_exec($conexion, $sql);
// Muestra los datos de la tabla luego del ingreso del registro
$result = pg_exec("SELECT codigo_producto, descripcion_producto FROM producto
order by codigo_producto");
$filas = pg_num_rows($result);
for($i=0;$i<$filas;$i++){
$dato1 = pg_result($result,$i,0);
$dato2 = pg_result($result,$i,1);
if(ceil($Contador/2) == ($Contador/2)){
echo "<tr bgcolor='#FEF3FE' class='TextoTabla'>
<td width='95'><div align='left'><a
href='Manten_Productos.php?CedCli=$dato1'>$dato1</a></div></td>
<td width='220'><div align='left'>$dato2</div></td>
</tr>";
}else{
echo "<tr bgcolor='#FFFFFF' class='TextoTabla'>

```

```

<td width='95' ><div align='left'><a
href='Manten_Productos.php?CedCli=$dato1'>$dato1 </a></div></td>
<td width='220'><div align='left'>$dato2</div></td>
</tr>";
}
$Contador++;
}
}else{
"<p align='center' class='TextoTabla1'>Funcionaria ya existe: Datos no se grabaron</p>";
}
}
//Para Modificar registros
if($valor1==2)
{
$CodigoProd = $_POST[txt_CodigoProd];
$DescripcionProd = $_POST[txt_DescripcionProd];
$StockMax = $_POST[txt_StockMax];
$StockMin = $_POST[txt_StockMin];
$Existencia = $_POST[txt_Existencia];
$TUnidad = $_POST[cmb_TUnidad];
$TPProducto = $_POST[cmb_TPProducto];
$Contador = 1;
// Modifica los datos a la tabla
$sql="UPDATE producto SET descripcion_producto = '$DescripcionProd', stock_maximo
= $StockMax, stock_minimo = $StockMin, existencia = $Existencia, codigo_tunidad =
'$TUnidad', codigo_tproducto = $TPProducto WHERE codigo_producto = '$CodigoProd'";
pg_exec($conexion, $sql);
// Muestra los datos de la tabla luego del ingreso del registro
$result = pg_exec("SELECT codigo_producto, descripcion_producto FROM producto
order by codigo_producto");
$filas = pg_num_rows($result);
for($i=0;$i<$filas;$i++){
$dato1 = pg_result($result,$i,0);
$dato2 = pg_result($result,$i,1);
if(ceil($Contador/2) == ($Contador/2)){
echo "<tr bgcolor='#FEF3FE' class='TextoTabla'>
<td width='95' ><div align='left'><a
href='Manten_Productos.php?CedCli=$dato1'>$dato1 </a></div></td>
<td width='220'><div align='left'>$dato2</div></td>
</tr>";
}
}
$Contador++;
}
}
//Para ver registros
if($valor1==3)
{
$Contador = 1;
// Muestra los datos de la tabla luego del ingreso del registro
$result = pg_exec("SELECT codigo_producto, descripcion_producto FROM producto
order by codigo_producto");
$filas = pg_num_rows($result);
for($i=0;$i<$filas;$i++){
$dato1 = pg_result($result,$i,0);
$dato2 = pg_result($result,$i,1);
if(ceil($Contador/2) == ($Contador/2)){
echo "<tr bgcolor='#FEF3FE' class='TextoTabla'>
<td width='95' ><div align='left'><a
href='Manten_Productos.php?CedCli=$dato1'>$dato1 </a></div></td>

```

```

        <td width='220'><div align='left'>$dato2</div></td>
    </tr>";
    }else{
    echo "<tr bgcolor='#FFFFFF' class='TextoTabla'>
    <td width='95'><div align='left'><a
    href='Manten_Productos.php?CedCli=$dato1'>$dato1</a></div></td>
    <td width='220'><div align='left'>$dato2</div></td>
    </tr>";
    }
    $Contador++;
    }
    pg_FreeResult($result);
    ?>
</table>
<table width="200" border="0" cellpadding="0" cellspacing="0">
    <tr>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td><div align="center">
            <input name="btn_Volver" type="submit" id="btn_Volver" value="Volver">
        </div>
        <div align="center"></div></td>
    </tr>
</table>
<p><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif"> </font></p>
</div>
</form>
<div align="justify"></div></TD>
</TR>
</TBODY>
</TABLE>
<DIV align=center></DIV></TD>
</TR>
</TBODY>
</TABLE>
</body>
</html>

```

#### Manten\_Producto.php

```

<html>
<head>
<title>Productos - SISCON</title>
</head>
<body>
<?php
    $CodigoProd1 = $_HTTP_GET_VARS['CedCli'];
    $Contador = 1;
    include 'Conexion.php';
    //Hace una consulta para ver datos de la habitacion
    $result=pg_exec("SELECT * FROM producto where codigo_producto = '$CodigoProd1'");
    $filas=pg_numrows($result);
    ?>
    <TABLE width=592 height=259
        border=0 align=center cellPadding=0 cellSpacing=0>
    <TBODY>
    <TR>
        <TD width=501 height="259" vAlign=top><TABLE height=64 cellSpacing=0 cellPadding=0 width=725
            align=center border=0>
        <TBODY>
        <TR>
            <TD width=725 height=64><form action="Ingreso_Productos.php?valor=2" method="post"
                name="frm_UsuarioFinal2" id="frm_UsuarioFinal2">

```

```

<p align="center"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif"><strong>PRODUCTOS</strong></font></p>
<div align="center">
  <table width="636" border="0" cellpadding="0" cellspacing="0">
    <tr>
      <td width="160"><div align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">CODIGO:</font></div></td>
      <td width="476"><div align="left"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif"><?php echo "$CodigoProd1"; ?>
        <?php $CodigoProd1= pg_result($result, 0, 0);
          echo "<input name='txt_CodigoProd' type='hidden' id='txt_CodigoProd' value='$CodigoProd1'>";
        ?>
      </font></div></td>
    </tr>
    <tr>
      <td><p align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">DESCRIPCION:</font></p></td>
      <td>
        <div align="left">
          <?php $DescripcionProd1 = pg_result($result, 0, 6);
          echo "<input name='txt_DescripcionProd' type='text' id='txt_DescripcionProd' size='60' maxlength='100' value='$DescripcionProd1'>";
          ?>
        </div>
      </td>
    </tr>
    <tr>
      <td><div align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">STOCK MAXIMO:</font></div></td>
      <td><div align="left"><?php $StockMax1 = pg_result($result, 0, 2);
        echo "<input name='txt_StockMax' type='text' id='txt_StockMax' size='20' maxlength='15' value='$StockMax1'>";
        ?>
      </div></td>
    </tr>
    <tr>
      <td><div align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">STOCK MINIMO:</font></div></td>
      <td><div align="left"><?php $StockMin1 = pg_result($result, 0, 1);
        echo "<input name='txt_StockMin' type='text' id='txt_StockMin' size='20' maxlength='15' value='$StockMin1'>";
        ?>
      </div></td>
    </tr>
    <tr>
      <td><div align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">EXISTENCIA:</font></div></td>
      <td><div align="left"><?php $Existencia1 = pg_result($result, 0, 3);
        echo "<input name='txt_Existencia' type='text' id='txt_Existencia' size='20' maxlength='15' value='$Existencia1'>";
        ?>
      </div></td>
    </tr>
    <tr>
      <td><div align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">TIPO DE UNIDAD:</font></div></td>
      <td><div align="left"><?php /*$TUnidad1 = pg_result($result, 0, 4);
        echo "<input name='txt_TUnidad' type='text' id='txt_TUnidad' size='50' maxlength='100' value='$TUnidad1'>";*/
        ?>
        <?php
        $TUnidad1 = pg_result($result, 0, 4);
        $result0 = pg_exec("SELECT codigo_tunidad, descripcion_tunidad
        FROM tipounidad
        order by codigo_tunidad");
        $filas0 = pg_num_rows($result0);
        $BanderaTU = 0;
        echo "<select name='cmb_TUnidad'>";
        for($i=0;$i<$filas0;$i++)

```

```

{
    $dato1 = pg_result($result0,$i,0);
    $dato2 = pg_result($result0,$i,1);
    if($TUnidad1 == $dato1){
        $BanderaTU = 1;
    }else{
        $BanderaTU = 0;
    }
    if ($BanderaTU == 1){
        echo "<option value='$dato1' selected>$dato2</option>";
    }else{
        echo "<option value='$dato1'>$dato2</option>";
    }
}
echo "</select>";
?>
</div></td>
</tr>
<tr>
    <td><div align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">TIPO DE PRODUCTO:</font></div></td>
    <td><div align="left">
        <?php /*$TProducto1 = pg_result($result, 0, 5);
            echo "<input name='txt_TProducto' type='text' id='txt_TProducto' size='50'
                maxlength='100' value='$TProducto1'>";*/
        ?>
    <?php
        $TProducto1 = pg_result($result, 0, 5);
        $result1 = pg_exec("SELECT codigo_tproducto, descripcion_tproducto
        FROM tipoproducto
        order by codigo_tproducto");
        $filas1 = pg_num_rows($result1);
        $BanderaTP = 0;
        echo "<select name='cmb_TProducto'>";
        for($i=0;$i<$filas1;$i++)
        {
            $dato1_1 = pg_result($result1,$i,0);
            $dato2_1 = pg_result($result1,$i,1);
            if($TProducto1 == $dato1_1){
                $BanderaTP = 1;
            }else{
                $BanderaTP = 0;
            }
            if ($BanderaTP == 1){
                echo "<option value='$dato1_1' selected>$dato2_1</option>";
            }else{
                echo "<option value='$dato1_1'>$dato2_1</option>";
            }
        }
        echo "</select>";
        ?>
    </div></td>
</tr>
</table>
<?php
    pg_FreeResult($result);
    ?>
<table width="260" border="0" cellpadding="0" cellspacing="0">
    <tr>
        <td width="260">&nbsp;</td>
    </tr>
    <tr>
        <td><div align="center">
            <input name="btn_Grabar" type="submit" id="btn_Grabar" value="Guardar Cambios">
        </div></td>
    </tr>
    </table>

```







```

if(ceil($Contador/2) == ($Contador/2)){
echo "<tr bgcolor='#FEF3FE' class='TextoTabla'>
<td width='95' ><div align='left'><a
href='Manten_TipoUnidad.php?CedCli=$dato1'>$dato1 </a></div></td>
<td width='220'><div align='left'>$dato2</div></td>
</tr>";
}else{
echo "<tr bgcolor='#FFFFFF' class='TextoTabla'>
<td width='95' ><div align='left'><a
href='Manten_TipoUnidad.php?CedCli=$dato1'>$dato1 </a></div></td>
<td width='220'><div align='left'>$dato2</div></td>
</tr>";
}
$Contador++;
}
}
pg_FreeResult($result);
?>
</table>
<table width="200" border="0" cellpadding="0" cellspacing="0">
<tr>
<td>&nbsp;</td>
</tr>
<tr>
<td><div align="center">
<input name="btn_Volver" type="submit" id="btn_Volver" value="Volver">
</div>
<div align="center"></div></td>
</tr>
</table>
<p>&nbsp;</p>
</div>
</form>
</TD>
</TR>
</TBODY>
</TABLE>
</body>
</html>

```

### Manten\_TipoUnidad

```

<html>
<head>
<title>Usuario Final - SISCON</title>
</head>
<body>
<?php
    $CodigoTU1 = $_HTTP_GET_VARS['CedCli'];
    $Contador = 1;
    //Establece la conexión con mysql
    include 'Conexion.php';
    //Hace una consulta para ver datos de la habitacion
    $result=pg_exec("SELECT * FROM tipounidad where codigo_tunidad = '$CodigoTU1'");
    $filas=pg_numrows($result);
?>
<TABLE width=592 height=259
    border=0 align=center cellPadding=0 cellSpacing=0>
<TBODY>
<TR>
<TD width=501 height="259" vAlign=top><TABLE height=64 cellSpacing=0 cellPadding=0 width=725
    align=center border=0>
<TBODY>
<TR>

```

```

<TD width=725 height=64><form action="Ingreso_TipoUnidad.php?valor=2" method="post"
name="frm_UsuarioFinal2" id="frm_UsuarioFinal2">
<p align="center"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
serif"><strong>TIPO DE UNIDAD </strong></font></p>
<div align="center">
<table width="636" border="0" cellpadding="0" cellspacing="0">
<tr>
<td width="160"><div align="left"><font color="#002E5B" size="2" face="Verdana, Arial,
Helvetica, sans-serif">CODIGO:</font></div></td>
<td width="476"><div align="left"><font color="#002E5B" size="2" face="Verdana, Arial,
Helvetica, sans-serif"><?php echo "$CodigoTU1"; ?>
<?php $CodigoTU1= pg_result($result, 0, 0);
echo "<input name='txt_CodigoTU' type='hidden' id='txt_CodigoTU' value='$CodigoTU1'>";
?>
</font></div></td></tr>
<tr>
<td><p align="left"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
serif">DESCRIPCION:</font></p></td>
<td><div align="left">
<?php $DescripcionTU1 = pg_result($result, 0, 1);
echo "<input name='txt_DescripcionTU' type='text' id='txt_DescripcionTU' size='55'
maxlength='50' value='$DescripcionTU1'>";
?>
</div></td>
</tr>
</table>
<?php
pg_FreeResult($result);
?>
<table width="260" border="0" cellpadding="0" cellspacing="0">
<tr>
<td width="260">&nbsp;</td>
</tr>
<tr>
<td><div align="center">
<input name="btn_Grabar" type="submit" id="btn_Grabar" value="Guardar Cambios">
</div></td></tr>
</table>
<p><font size="2" face="Verdana, Arial, Helvetica, sans-serif"><a
href="Ingreso_TipoUnidad.php?valor=3"> Ver Tipos de Unidad </a></font></p>
</div> </form>
</TD>
</TR>
</TBODY>
</TABLE></TD>
</TR>
</TBODY>
</TABLE>
</body>
</html>

```

#### Ingreso\_TipoProducto.php

```

<html>
<head>
<title>Tipo Producto - SISCON</title>
</head>
<body>
<TABLE height=284 cellSpacing=0 cellPadding=0 width=592 align=center border=0>
<TBODY>
<TR>
<TD vAlign=top width=91 >&nbsp;</TD>
<TD vAlign=top width=501><TABLE height=64 cellSpacing=0 cellPadding=0 width=488 align=center
border=0>
<TBODY>

```

```

<TR>
<TD width=468 height=64><form action="TipoProducto.html" method="post" name="frm_Cliente1"
id="frm_Cliente1">
<p align="center"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
serif"><strong>TIPOS DE PRODUCTO </strong></font></p>
<div align="center">
<table width="440" border="1" cellpadding="0" cellspacing="0">
<tr bordercolor="#002E5B" bgcolor="#DCE0F1">
<td width="105"><div align="center"><strong><font color="#002E5B" size="2" face="Verdana,
Arial, Helvetica, sans-serif">CODIGO</font></strong></div></td>
<td width="329" bgcolor="#DCE0F1"><div align="center"><strong><font color="#002E5B"
size="2" face="Verdana, Arial, Helvetica, sans-serif">DESCRIPCION</font></strong></div></td>
</tr>
<?php
include 'Conexion.php';
//Para Insertar Registros
$valor1=$_HTTP_GET_VARS['valor'];
if($valor1==1)
{
$DescripcionTP = $_POST[txt_DescripcionTP];
$v_codigotp = pg_exec("select nextval('seq_cod_tpro')");
$filaSec = pg_numrows($v_codigotp);
$v_codigotp1 = pg_result($v_codigotp,0,0);
$Contador = 1;
//Hace una consulta para ver si cliente existe
$result=pg_exec("SELECT * FROM tipoproducto where codigo_tproducto =
$v_codigotp1");
//Ve el número de filas que obtiene el select si es 0 el cliente no existe
$filas=pg_numrows($result);
if ($filas==0)
{
// Inserta los datos a la tabla
$sql="INSERT INTO tipoproducto (codigo_tproducto, descripcion_tproducto) VALUES
($v_codigotp1,$DescripcionTP)";
pg_exec($conexion, $sql);
// Muestra los datos de la tabla luego del ingreso del registro
$result = pg_exec("SELECT codigo_tproducto, descripcion_tproducto FROM tipoproducto
order by codigo_tproducto");
$filas = pg_num_rows($result);
for($i=0;$i<$filas;$i++){
$dato1 = pg_result($result,$i,0);
$dato2 = pg_result($result,$i,1);
if(ceil($Contador/2) == ($Contador/2))
{
echo "<tr bgcolor='#FEF3FE' class='TextoTabla'>
<td width='95' ><div align='left'><a href='Manten_TipoProducto.php?CedCli=$dato1'>
$dato1</a></div></td>
<td width='220'><div align='left'>$dato2</div></td>
</tr>";
}
}
}
else
{
echo "<tr bgcolor='#FFFFFF' class='TextoTabla'>
<td width='95' ><div align='left'><a
href='Manten_TipoProducto.php?CedCli=$dato1'>$dato1</a></div></td>
<td width='220'><div align='left'>$dato2</div></td>
</tr>";
}
}
$Contador++;
} }
else
{
echo "<p align='center' class='TextoTabla1'>Tipo de Producto ya existe: Datos no se
grabaron</p>";
}
}

```

```

}}
//Para Modificar registros
if($valor1==2)
{
$CodigoTP = $_POST[txt_CodigoTP];
$DescripcionTP = $_POST[txt_DescripcionTP];
$Contador = 1;
// Modifica los datos a la tabla
$sql="UPDATE tipoproducto SET descripcion_tproducto = '$DescripcionTP' WHERE
codigo_tproducto = $CodigoTP";
pg_exec($conexion, $sql);
// Muestra los datos de la tabla luego del ingreso del registro
$result = pg_exec("SELECT codigo_tproducto, descripcion_tproducto FROM tipoproducto order by
codigo_tproducto");
$filas = pg_num_rows($result);
for($i=0;$i<$filas;$i++){
$dato1 = pg_result($result,$i,0);
$dato2 = pg_result($result,$i,1);
if(ceil($Contador/2) == ($Contador/2))
{
echo "<tr bgcolor='#FEF3FE' class='TextoTabla'>
<td width='95' ><div align='left'><a href='Manten_TipoProducto.php?CedCli=$dato1'>
$dato1</a></div></td>
<td width='220'><div align='left'>$dato2</div></td>
</tr>";
}
else
{
echo "<tr bgcolor='#FFFFFF' class='TextoTabla'>
<td width='95' ><div align='left'><a
href='Manten_TipoProducto.php?CedCli=$dato1'>$dato1</a></div></td>
<td width='220'><div align='left'>$dato2</div></td>
</tr>";
}
}
$Contador++;
} }
//Para ver registros
if($valor1==3)
{
$DescripcionTP = $_POST[txt_DescripcionTP];
$Contador = 1;
// Muestra los datos de la tabla luego del ingreso del registro
$result = pg_exec("SELECT codigo_tproducto, descripcion_tproducto FROM tipoproducto order by
codigo_tproducto");
$filas = pg_num_rows($result);
for($i=0;$i<$filas;$i++){
$dato1 = pg_result($result,$i,0);
$dato2 = pg_result($result,$i,1);
if(ceil($Contador/2) == ($Contador/2))
{
echo "<tr bgcolor='#FEF3FE' class='TextoTabla'>
<td width='95' ><div align='left'><a
href='Manten_TipoProducto.php?CedCli=$dato1'>$dato1</a></div></td>
<td width='220'><div align='left'>$dato2</div></td>
</tr>";
}
else
{
echo "<tr bgcolor='#FFFFFF' class='TextoTabla'>
<td width='95' ><div align='left'><a
href='Manten_TipoProducto.php?CedCli=$dato1'>$dato1</a></div></td>
<td width='220'><div align='left'>$dato2</div></td>
</tr>";
}
}
}

```

## Manten\_TipoProducto.php

```
<html>
<head>
<title>Usuario Final - SISCON</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body>
<?php
    $CodigoTP1 = $_HTTP_GET_VARS['CedCli'];
    $Contador = 1;
    //Establece la conexión con mysql
    include 'Conexion.php';
    //Hace una consulta para ver datos de la habitacion
    $result=pg_exec("SELECT * FROM tipoproducto where codigo_tproducto = $CodigoTP1");
    $filas=pg_numrows($result);
?>
<TABLE width=592 height=284 border=0 align=center cellPadding=0 cellSpacing=0>
<TBODY>
<TR>
<TD vAlign=top width=501><TABLE height=64 cellSpacing=0 cellPadding=0 width=488 align=center border=0>
<TBODY>
<TR>
<TD width=468 height=64><form action="Ingreso_TipoProducto.php?valor=2" method="post" name="frm_UsuarioFinal2" id="frm_UsuarioFinal2">
<p align="center"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif"><strong>TIPOS DE PRODUCTO </strong></font></p>
<div align="center">
<table width="424" border="0" cellpadding="0" cellspacing="0">
<tr>
<td width="112"><div align="left"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">CODIGO:</font></div></td>
<td width="312"><div align="left"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif"><?php echo "$CodigoTP1"; ?>
<?php $CodigoTP1= pg_result($result, 0, 0);
echo "<input name='txt_CodigoTP' type='hidden' id='txt_CodigoTP' value='$CodigoTP1'>";
?>
</font></div></td>
```

```

        </tr> <tr>
        <td><p align="left"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
        serif">DESCRIPCION:</font></p></td>
        <td><div align="left">
        <?php $DescripcionTP1 = pg_result($result, 0, 1);
        echo "<input name='txt_DescripcionTP' type='text' id='txt_DescripcionTP' size='55'
        maxlength='50' value='$DescripcionTP1'>";
        ?>
        </div></td>
        </tr>
    </table>
    <?php
        pg_FreeResult($result);
        ?>
    <table width="260" border="0" cellpadding="0" cellspacing="0">
    <tr>
    <td width="260">&nbsp;</td>
    </tr>
    <tr>
    <td><div align="center">
    <input name="btn_Grabar" type="submit" id="btn_Grabar" value="Guardar Cambios">
    </div></td> </tr>
    </table>
    <p><font size="2" face="Verdana, Arial, Helvetica, sans-serif"><a
    href="Ingreso_TipoProducto.php?valor=3">Ver Tipos de Producto </a></font></p>
    </div>
    <p align="center">&nbsp;</p>
    </form>
    <div align="justify"></div></TD>
    </TR>
    </TBODY> </TABLE></TD>
    </TR>
    </TBODY>
    </TABLE>
    </body>
    </html>

```

### Ingreso\_Proveedores.php

```

<html>
<head>
<title>Proveedores - SISCON</title>
</head>
<body>
<TABLE height=64 cellSpacing=0 cellPadding=0 width=474 align=center border=0>
    <TBODY>
    <TR>
    <TD width=474 height=64><form action="Proveedores.html" method="post" name="frm_Funcionaria2"
    id="frm_Funcionaria2">
    <p align="center"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
    serif"><strong>PROVEEDORES</strong></font></p>
    <div align="center">
    <table width="440" border="1" cellpadding="0" cellspacing="0">
    <tr bordercolor="#002E5B" bgcolor="#DCE0F1">
    <td width="105"><div align="center"><strong><font color="#002E5B" size="2" face="Verdana, Arial,
    Helvetica, sans-serif">RUC</font></strong></div></td>
    <td width="329" bgcolor="#DCE0F1"><div align="center"><strong><font color="#002E5B" size="2"
    face="Verdana, Arial, Helvetica, sans-serif">DESCRIPCION</font></strong></div></td>
    </tr>
    </table>
    <?php
        include 'Conexion.php';
        //Para Insertar Registros
        valor1=$HTTP_GET_VARS['valor'];
        if($valor1==1)
        {

```



```

$RUC = $_POST[txt_RUC];
$RazonSocialPro = $_POST[txt_RazonSocialPro];
$DireccionPro = $_POST[txt_DireccionPro];
$TelefonoPro = $_POST[txt_TelefonoPro];
$v_codigopro = pg_exec("select nextval('seq_cod_pro')");
$filaSec = pg_numrows($v_codigopro);
$v_codigopro1 = pg_result($v_codigopro,0,0);
$Contador = 1;
//Hace una consulta para ver si cliente existe
$result=pg_exec("SELECT * FROM proveedor where codigo_proveedor = $v_codigopro1");
//Ve el número de filas que obtiene el select si es 0 el cliente no existe
$filas=pg_numrows($result);
if($filas == 0){
// Inserta los datos a la tabla
$sql="INSERT INTO proveedor (codigo_proveedor, ruc, razonsocial_proveedor,
direccion_proveedor, telefono_proveedor) VALUES
($v_codigopro1,$RUC,$RazonSocialPro,$DireccionPro,$TelefonoPro)";
pg_exec($conexion, $sql);
// Muestra los datos de la tabla luego del ingreso del registro
$result = pg_exec("SELECT ruc, razonsocial_proveedor FROM proveedor order by ruc");
$filas = pg_num_rows($result);
for($i=0;$i<$filas;$i++){
$dato1 = pg_result($result,$i,0);
$dato2 = pg_result($result,$i,1);
if(ceil($Contador/2) == ($Contador/2)){
echo "<tr bgcolor=#FEF3FE class='TextoTabla'>
<td width='95'><div align='left'><a
href='Manten_Proveedores.php?CedCli=$dato1'>$dato1</a></div></td>
<td width='220'><div align='left'>$dato2</div></td>
</tr>";
}else{
echo "<tr bgcolor='#FFFFFF' class='TextoTabla'>
<td width='95'><div align='left'><a href='Manten_Proveedores.php?CedCli=$dato1'>
$dato1</a></div></td>
<td width='220'><div align='left'>$dato2</div></td>
</tr>";
}
}
$Contador++;
}
}else{
echo "<p align='center' class='TextoTabla1'>Proveedor ya existe: Datos no se
grabaron</p>";
}}
//Para Modificar registros
if($valor1==2)
{
$RUC = $_POST[txt_RUC];
$RazonSocialPro = $_POST[txt_RazonSocialPro];
$DireccionPro = $_POST[txt_DireccionPro];
$TelefonoPro = $_POST[txt_TelefonoPro];
$Contador = 1;
// Modifica los datos a la tabla
$sql="UPDATE proveedor SET razonsocial_proveedor = '$RazonSocialPro',
direccion_proveedor = '$DireccionPro', telefono_proveedor = '$TelefonoPro' WHERE ruc =
'$RUC'";
pg_exec($conexion, $sql);
// Muestra los datos de la tabla luego del ingreso del registro
$result = pg_exec("SELECT ruc, razonsocial_proveedor FROM proveedor order by ruc");
$filas = pg_num_rows($result);
for($i=0;$i<$filas;$i++){
$dato1 = pg_result($result,$i,0);
$dato2 = pg_result($result,$i,1);
if(ceil($Contador/2) == ($Contador/2)){
echo "<tr bgcolor=#FEF3FE class='TextoTabla'>

```

```

        <td width='95' ><div align='left'><a href='Manten_Proveedores.php?CedCli=$dato1'>
        $dato1 </a></div></td>
        <td width='220'><div align='left'>$dato2</div></td>
        </tr>";
    }else{
        echo "<tr bgcolor='#FFFFFF' class='TextoTabla'>
        <td width='95' ><div align='left'><a
        href='Manten_Proveedores.php?CedCli=$dato1'>$dato1 </a></div></td>
        <td width='220'><div align='left'>$dato2</div></td>
        </tr>";
    }
    $Contador++;
}
}
//Para ver registros
if($valor1==3)
{
    $Contador = 1;
    // Muestra los datos de la tabla luego del ingreso del registro
    $result = pg_exec("SELECT ruc, razonsocial_proveedor FROM proveedor order by ruc");
    filas = pg_num_rows($result);
    for($i=0;$i<$filas;$i++){
        $dato1 = pg_result($result,$i,0);
        $dato2 = pg_result($result,$i,1);
        if(ceil($Contador/2) == ($Contador/2)){
            echo "<tr bgcolor='#FEF3FE' class='TextoTabla'>
            <td width='95' ><div align='left'><a href='Manten_Proveedores.php?CedCli=$dato1'>
            $dato1 </a></div></td>
            <td width='220'><div align='left'>$dato2</div></td>
            </tr>";
        }else{
            echo "<tr bgcolor='#FFFFFF' class='TextoTabla'>
            <td width='95' ><div align='left'><a href='Manten_Proveedores.php?CedCli=$dato1'>
            $dato1 </a></div></td>
            <td width='220'><div align='left'>$dato2</div></td>
            </tr>";
        }
        $Contador++;
    }
    pg_FreeResult($result);
?>
</table>
<table width="200" border="0" cellpadding="0" cellspacing="0">
    <tr>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td><div align="center">
            <input name="btn_Volver" type="submit" id="btn_Volver" value="Volver">
        </div>
        <div align="center"></div></td>
    </tr>
</table>
<p>&nbsp;</p>
</div>
</form>
</TD>
</TR> </TBODY></TABLE>
</body>
</html>

```

**Manten\_Proveedores.php**

```

<html>
<head>

```

```

<title>Proveedores - SISCON</title>
</head>
<body>
<?php
    $CodigoPro1 = $_HTTP_GET_VARS['CedCli'];
    $Contador = 1;
    include 'Conexion.php';
    //Hace una consulta para ver datos de la habitacion
    $result=pg_exec("SELECT * FROM proveedor where ruc = '$CodigoPro1'");
    $filas=pg_numrows($result);
?>
<TABLE height=204 cellSpacing=0 cellPadding=0 width=725 align=left
border=0>
<TBODY>
<TR>
<TD width=725 height=204><form action="Ingreso_Proveedores.php?valor=2" method="post"
name="frm_UsuarioFinal2" id="frm_UsuarioFinal2">
<p align="center"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
serif"><strong>PROVEEDORES</strong></font></p>
<div align="center">
<table width="636" border="0" cellpadding="0" cellspacing="0">
<tr>
<td width="160"><div align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica,
sans-serif">RUC:</font></div></td>
<td width="476"><div align="left"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica,
sans-serif"><?php echo "$CodigoPro1"; ?>
<?php $CodigoPro1= pg_result($result, 0, 1);
echo "<input name='txt_RUC' type='hidden' id='txt_RUC' value='$CodigoPro1'>";
?>
</font></div></td></tr>
<tr>
<td><p align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
serif">RAZON SOCIAL:</font></p></td>
<td><div align="left">
<?php $RazonSocialPro1 = pg_result($result, 0, 2);
echo "<input name='txt_RazonSocialPro' type='text' id='txt_RazonSocialPro' size='65'
maxlength='100' value='$RazonSocialPro1'>";
?>
</div></td></tr>
<tr>
<td><div align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
serif">DIRECCION:</font></div></td>
<td><?php $DireccionPro1 = pg_result($result, 0, 3);
echo "<input name='txt_DireccionPro' type='text' id='txt_DireccionPro' size='65' maxlength='70'
value='$DireccionPro1'>";
?></td></tr>
<tr>
<td><div align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
serif">TELEFONO:</font></div></td>
<td><?php $TelefonoPro1 = pg_result($result, 0, 4);
echo "<input name='txt_TelefonoPro' type='text' id='txt_TelefonoPro' size='25' maxlength='20'
value='$TelefonoPro1'>";
?></td></tr>
</table>
<?php
    pg_FreeResult($result);
?>
<table width="260" border="0" cellpadding="0" cellspacing="0">
<tr>
<td width="260">&nbsp;</td>
</tr>
<tr>
<td>
<div align="center">
<input name="btn_Grabar" type="submit" id="btn_Grabar" value="Guardar Cambios">

```

```

        </div></td></tr>
    </table>
</div>
<p align="center"><font size="2" face="Verdana, Arial, Helvetica, sans-serif"><a
href="Ingreso_Proveedores.php?valor=3">Ver Proveedores </a></font></p>
</form></TD>
</TR> </TBODY></TABLE>
</body>
</html>

```

### Ingreso\_DestinoGeografico.php

```

<html>
<head>
<title>Destino Geogr&aacute;fico - SISCON</title>
</head>
<body>
<TABLE height=284 cellSpacing=0 cellPadding=0 width=592 align=center
border=0>
<TBODY>
<TR>
<TD vAlign=top width=91 >&nbsp;</TD>
<TD vAlign=top width=501><TABLE height=64 cellSpacing=0 cellPadding=0 width=488 align=center
border=0>
<TBODY>
<TR>
<TD width=468 height=64><form action="DestinoGeografico.html" method="post"
name="frm_Cliente1" id="frm_Cliente1">
<p align="center"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
serif"><strong>DESTINOS GEOGRAFICOS</strong></font></p>
<div align="center">
<table width="440" border="1" cellpadding="0" cellspacing="0">
<tr bordercolor="#002E5B" bgcolor="#DCE0F1">
<td width="105"><div align="center"><strong><font color="#002E5B" size="2" face="Verdana,
Arial, Helvetica, sans-serif">CODIGO</font></strong></div></td>
<td width="329" bgcolor="#DCE0F1"><div align="center"><strong><font color="#002E5B"
size="2" face="Verdana, Arial, Helvetica, sans-serif">DESCRIPCION</font></strong></div></td>
</tr>
</table>
<?php
include 'Conexion.php';
//Para Insertar Registros
$valor1=$_HTTP_GET_VARS['valor'];
if($valor1==1)
{
$CodigoUb = $_POST[txt_CodigoUb];
$DescripcionUb = $_POST[txt_DescripcionUb];
v_codigodg = pg_exec("select nextval('seq_cod_dgeo')");
$filaSec = pg_numrows($v_codigodg);
$v_codigodg1 = pg_result($v_codigodg,0,0);
$Contador = 1;
//Hace una consulta para ver si cliente existe
$result=pg_exec("SELECT * FROM destinogeo where codigo_ubicacion =
$v_codigodg1");
//Ve el n&uacute;mero de filas que obtiene el select si es 0 el cliente no existe
$filas=pg_numrows($result);
if ($filas==0)
{
// Inserta los datos a la tabla
$sql="INSERT INTO destinogeo (codigo_ubicacion, descripcion_ubicacion) VALUES
($v_codigodg1,$DescripcionUb)";
pg_exec($conexion, $sql);
// Muestra los datos de la tabla luego del ingreso del registro
$result = pg_exec("SELECT codigo_ubicacion, descripcion_ubicacion FROM destinogeo
order by codigo_ubicacion");
$filas = pg_num_rows($result);

```

```

for($i=0;$i<$filas;$i++){
    $dato1 = pg_result($result,$i,0);
    $dato2 = pg_result($result,$i,1);
    if(ceil($Contador/2) == ($Contador/2))
    {
        echo "<tr bgcolor='#FEF3FE' class='TextoTabla'>
        <td width='95' ><div align='left'><a href='Manten_DestinoGeografico.php?CedCli=$dato1'>
        $dato1</a></div></td>
        <td width='220'><div align='left'>$dato2</div></td>
        </tr>";
    }
    else
    {
        echo "<tr bgcolor='#FFFFFF' class='TextoTabla'>
        <td width='95' ><div align='left'><a href='Manten_DestinoGeografico.php?CedCli=$dato1'>
        $dato1</a></div></td>
        <td width='220'><div align='left'>$dato2</div></td>
        </tr>";
    }
    $Contador++;
} }
else
{
    echo "<p align='center' class='TextoTabla1'>Destino Geografico ya existe: Datos no se
    grabaron</p>";
}
}
//Para Modificar registros
if($valor1==2)
{
    $CodigoUb = $_POST[txt_CodigoUb];
    $DescripcionUb = $_POST[txt_DescripcionUb];
    $Contador = 1;
    // Modifica los datos a la tabla
    $sql ="UPDATE destinogeo SET descripcion_ubicacion = '$DescripcionUb' WHERE
    codigo_ubicacion = '$CodigoUb'";
    pg_exec($conexion, $sql);
    // Muestra los datos de la tabla luego del ingreso del registro
    $result = pg_exec("SELECT codigo_ubicacion, descripcion_ubicacion FROM destinogeo
    order by codigo_ubicacion");
    $filas = pg_num_rows($result);
    for($i=0;$i<$filas;$i++){
        $dato1 = pg_result($result,$i,0);
        $dato2 = pg_result($result,$i,1);
        if(ceil($Contador/2) == ($Contador/2))
        echo "<tr bgcolor='#FEF3FE' class='TextoTabla'>
        <td width='95' ><div align='left'><a href='Manten_DestinoGeografico.php?CedCli=$dato1'>
        $dato1</a></div></td>
        <td width='220'><div align='left'>$dato2</div></td>
        </tr>";
    }
    else
    {
        echo "<tr bgcolor='#FFFFFF' class='TextoTabla'>
        <td width='95' ><div align='left'><a href='Manten_DestinoGeografico.php?CedCli=$dato1'>
        $dato1</a></div></td>
        <td width='220'><div align='left'>$dato2</div></td>
        </tr>";
    }
    $Contador++;
} }
//Para ver registros
if($valor1==3)
{
    $CodigoUb = $_POST[txt_CodigoUb];

```

[illegible]

```

<TR>
<TD vAlign=top width=501><TABLE height=64 cellSpacing=0 cellPadding=0 width=488 align=center
border=0>
<TBODY>
<TR>
<TD width=468 height=64><form action="Ingreso_DestinoGeografico.php?valor=2" method="post"
name="frm_UsuarioFinal2" id="frm_UsuarioFinal2">
<p align="center"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
serif"><strong>DESTINOS GEOGRAFICOS</strong></font></p>
<div align="center">
<table width="424" border="0" cellpadding="0" cellspacing="0">
<tr>
<td width="112"><div align="left"><font color="#002E5B" size="2" face="Verdana, Arial,
Helvetica, sans-serif">CODIGO:</font></div></td>
<td width="312"><div align="left"><font color="#002E5B" size="2" face="Verdana, Arial,
Helvetica, sans-serif"><?php echo "$CodigoUb1"; ?>
<?php $CodigoUb1= pg_result($result, 0, 0);
echo "<input name='txt_CodigoUb' type='hidden' id='txt_CodigoUb' value='$CodigoUb1'>";
?>
</font></div></td></tr>
<tr>
<td><p align="left"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
serif">DESCRIPCION:</font></p></td>
<td><div align="left">
<?php $DescripcionUb1 = pg_result($result, 0, 1);
echo "<input name='txt_DescripcionUb' type='text' id='txt_DescripcionUb' size='65'
maxlength='100' value='$DescripcionUb1'>";
?>
</div></td></tr></table>
<?php
pg_FreeResult($result);
?>
<table width="260" border="0" cellpadding="0" cellspacing="0">
<tr>
<td width="260">&nbsp;</td></tr>
<tr>
<td><div align="center">
<input name="btn_Grabar" type="submit" id="btn_Grabar" value="Guardar Cambios">
</div></td></tr></table>
<p><font size="2" face="Verdana, Arial, Helvetica, sans-serif"><a
href="Ingreso_DestinoGeografico.php?valor=3">Ver Destinos Geogr&acute;ficos</a><a
href="Ingreso_TipoDocumento.php?valor=3"></a><a
href="Ingreso_UsuarioFinal.php?valor=3"></a></font></p>
</div>
</form>
<div align="justify"></div></TD>
</TR> </TBODY> </TABLE>
<DIV align=center></DIV></TD>
</TR> </TBODY></TABLE>
</body>
</html>

```

#### Ingreso\_Responsable.php

```

<html>
<head>
<title>Responsable - SISCON</title>
</head>
<body>
<TABLE height=64 cellSpacing=0 cellPadding=0 width=474 align=center border=0>
<TBODY>
<TR>
<TD width=474 height=64><form action="Responsable.html" method="post" name="frm_Funcionaria2"
id="frm_Funcionaria2">
<p align="center"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
serif"><strong>RESPONSABLE</strong></font></p>

```

```

<div align="center">
<table width="440" border="1" cellpadding="0" cellspacing="0">
<tr bordercolor="#002E5B" bgcolor="#DCE0F1">
<td width="105"><div align="center"><strong><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">CEDULA</font></strong></div></td>
<td width="329" bgcolor="#DCE0F1"><div align="center"><strong><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">APELLIDO NOMBRE</font></strong></div></td>
</tr>
<?php
include 'Conexion.php';
//Para Insertar Registros
$valor1=$_HTTP_GET_VARS['valor'];
if($valor1==1)
{
$CedulaRes = $_POST[txt_CedulaRes];
$NombreRes = $_POST[txt_NombreRes];
$ApellidoRes = $_POST[txt_ApellidoRes];
$TelefonoRes = $_POST[txt_TelefonoRes];
$v_codigores = pg_exec("select nextval('seq_cod_res')");
$filaSec = pg_numrows($v_codigores);
$v_codigores1 = pg_result($v_codigores,0,0);
$Contador = 1;
//Hace una consulta para ver si cliente existe
$result=pg_exec("SELECT * FROM responsable where codigo_responsable = $v_codigores1");
//Ve el número de filas que obtiene el select si es 0 el cliente no existe
$filas=pg_numrows($result);
if($filas == 0){
// Inserta los datos a la tabla
$sql ="INSERT INTO responsable (codigo_responsable, cedula_responsable, nombre_responsable, apellido_responsable, telefono_responsable) VALUES ($v_codigores1, '$CedulaRes', '$NombreRes','$ApellidoRes','$TelefonoRes')";
pg_exec($conexion, $sql);
// Muestra los datos de la tabla luego del ingreso del registro
$result = pg_exec("SELECT cedula_responsable, apellido_responsable, nombre_responsable FROM responsable order by cedula_responsable");
$filas = pg_num_rows($result);
for($i=0;$i<$filas;$i++){
$dato1 = pg_result($result,$i,0);
$dato2 = pg_result($result,$i,1);
$dato3 = pg_result($result,$i,2);
if(ceil($Contador/2) == ($Contador/2)){
echo "<tr bgcolor='#FEF3FE' class='TextoTabla'>
<td width='95' ><div align='left'><a href='Manten_Responsable.php?CedCli=$dato1'>
$dato1</a></div></td>
<td width='220'><div align='left'>$dato2 $dato3</div></td>
</tr>";
}else{
echo "<tr bgcolor='#FFFFFF' class='TextoTabla'>
<td width='95' ><div align='left'><a href='Manten_Responsable.php?CedCli=$dato1'>
$dato1</a></div></td>
<td width='220'><div align='left'>$dato2 $dato3</div></td>
</tr>";
}
$Contador++;
}
}else{
echo "<p align='center' class='TextoTabla1'>Responsable ya existe: Datos no se grabaron</p>";
}}
//Para Modificar registros
if($valor1==2)
{
$CodigoRes = $_POST[txt_CodigoRes];

```



```

$CedulaRes = $_POST[txt_CedulaRes];
$NombreRes = $_POST[txt_NombreRes];
$ApellidoRes = $_POST[txt_ApellidoRes];
$TelefonoRes = $_POST[txt_TelefonoRes];
$Contador = 1;
// Modifica los datos a la tabla
sql = "UPDATE responsable SET cedula_responsable = '$CedulaRes' ,
nombre_responsable = '$NombreRes', apellido_responsable = '$ApellidoRes',
telefono_responsable = '$TelefonoRes' WHERE codigo_responsable = $CodigoRes";
pg_exec($conexion, $sql);
// Muestra los datos de la tabla luego del ingreso del registro
$result = pg_exec("SELECT cedula_responsable, apellido_responsable,
nombre_responsable FROM responsable order by cedula_responsable");
$filas = pg_num_rows($result);
for($i=0;$i<$filas;$i++){
$dato1 = pg_result($result,$i,0);
$dato2 = pg_result($result,$i,1);
$dato3 = pg_result($result,$i,2);
if(ceil($Contador/2) == ($Contador/2)){
echo "<tr bgcolor='#FEF3FE' class='TextoTabla'>
<td width='95' ><div align='left'><a href='Manten_Responsable.php?CedCli=$dato1'>
$dato1</a></div></td>
<td width='220'><div align='left'>$dato2 $dato3</div></td>
</tr>";
}else{
echo "<tr bgcolor='#FFFFFF' class='TextoTabla'>
<td width='95' ><div align='left'><a href='Manten_Responsable.php?CedCli=$dato1'>
$dato1</a></div></td>
<td width='220'><div align='left'>$dato2 $dato3</div></td>
</tr>";
}
$Contador++;
}
}
//Para ver registros
if($valor1==3)
{
//echo "<h4>Diego $valor</h4>";
$Contador = 1;
// Muestra los datos de la tabla luego del ingreso del registro
$result = pg_exec("SELECT cedula_responsable, apellido_responsable,
nombre_responsable FROM responsable order by cedula_responsable");
$filas = pg_num_rows($result);
for($i=0;$i<$filas;$i++){
$dato1 = pg_result($result,$i,0);
$dato2 = pg_result($result,$i,1);
$dato3 = pg_result($result,$i,2);
if(ceil($Contador/2) == ($Contador/2)){
echo "<tr bgcolor='#FEF3FE' class='TextoTabla'>
<td width='95' ><div align='left'><a href='Manten_Responsable.php?CedCli=$dato1'>
$dato1</a></div></td>
<td width='220'><div align='left'>$dato2 $dato3</div></td>
</tr>";
}else{
echo "<tr bgcolor='#FFFFFF' class='TextoTabla'>
<td width='95' ><div align='left'><a href='Manten_Responsable.php?CedCli=$dato1'>
$dato1</a></div></td>
<td width='220'><div align='left'>$dato2 $dato3</div></td>
</tr>";
}
$Contador++;
}
}
pg_FreeResult($result);
?>
</table>

```

```

<table width="200" border="0" cellpadding="0" cellspacing="0">
  <tr>
    <td>&nbsp;  </td>
  </tr>
  <tr>
    <td><div align="center">
      <input name="btn_Volver" type="submit" id="btn_Volver" value="Volver">
    </div>
    <div align="center"></div></td></tr>
</table>
<p>&nbsp;</p>
</div>
</form>
</TD>
</TR> </TBODY></TABLE>
</body>
</html>

Manten_Responsable.php
<html>
<head>
<title>Responsable - SISCON</title>
</head>
<body>
<?php
  $CodigoRes1 = $_HTTP_GET_VARS['CedCli'];
  $Contador = 1;
  include 'Conexion.php';
  //Hace una consulta para ver datos de la habitacion
  $result=pg_exec("SELECT * FROM responsable where cedula_responsable = '$CodigoRes1'");
  $filas=pg_numrows($result);
?>
<TABLE height=204 cellSpacing=0 cellPadding=0 width=725 align=left border=0>
  <TBODY>
    <TR>
      <TD width=725 height=204><form action="Ingreso_Responsable.php?valor=2" method="post"
        name="frm_UsuarioFinal2" id="frm_UsuarioFinal2">
        <p align="center"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
          serif"><strong>RESPONSABLE</strong></font></p>
        <div align="center">
          <table width="636" border="0" cellpadding="0" cellspacing="0">
            <tr>
              <td width="160"><div align="right"><font color="#002E5B" size="2" face="Verdana, Arial,
                Helvetica, sans-serif">CODIGO:</font></div></td>
              <td width="476"><div align="left"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica,
                sans-serif">
                <?php $CodigoRes2= pg_result($result, 0, 0);
                echo "$CodigoRes2";
                echo "<input name='txt_CodigoRes' type='hidden' id='txt_CodigoRes'
                  value='$CodigoRes2'>";
                ?>
              </font></div></td></tr>
            <tr>
              <td><p align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
                serif">CEDULA:</font></p></td>
              <td><div align="left">
                <font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">
                <?php $CedulaRes= pg_result($result, 0, 1);
                echo "$CedulaRes";
                echo "<input name='txt_CedulaRes' type='hidden' id='txt_CedulaRes'
                  value='$CedulaRes'>";
                ?>
              </font> </div></td></tr>
            <tr>

```

```

        <td><div align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">NOMBRE:</font></div></td>
        <td><div align="left">
            <?php $NombreRes1 = pg_result($result, 0, 2);
            echo "<input name='txt_NombreRes' type='text' id='txt_NombreRes' size='30'
            maxlength='25' value='$NombreRes1'>";
        ?>
        </div></td></tr>
    <tr>
        <td><div align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">APELLIDO:</font></div></td>
        <td><div align="left">
            <?php $ApellidoRes1 = pg_result($result, 0, 3);
            echo "<input name='txt_ApellidoRes' type='text' id='txt_ApellidoRes' size='30'
            maxlength='25' value='$ApellidoRes1'>";
        ?>
        </div></td></tr>
    <tr>
        <td><div align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">TELEFONO:</font></div></td>
        <td><div align="left">
            <?php $TelefonoRes1 = pg_result($result, 0, 4);
            echo "<input name='txt_TelefonoRes' type='text' id='txt_TelefonoRes' size='25'
            maxlength='20' value='$TelefonoRes1'>";
        ?>
        </div></td></tr>
    </table>
    <?php
        pg_FreeResult($result);
    ?>
    <table width="260" border="0" cellpadding="0" cellspacing="0">
    <tr>
    <td width="260">&nbsp;</td>
    </tr>
    <tr>
    <td><div align="center">
        <input name="btn_Grabar" type="submit" id="btn_Grabar" value="Guardar Cambios">
    </div></td></tr>
    </table>
    </div>
    <p align="center"><font size="2" face="Verdana, Arial, Helvetica, sans-serif"><a
    href="Ingreso_Responsable.php?valor=3">Ver Responsables </a></font></p>
    </form></TD>
    </TR> </TBODY></TABLE>
</body>
</html>

```

### Ingreso\_Organizacion.php

```

<html>
<head>
<title>Organizaciones - SISCON</title>
</head>
<body>
<TABLE height=64 cellSpacing=0 cellPadding=0 width=474 align=center border=0>
<TBODY>
    <TR>
    <TD width=474 height=64><form action="organizacion.html" method="post" name="frm_Funcionaria2"
    d="frm_Funcionaria2">
    <p align="center"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
    erif"><strong>ORGANIZACIONES</strong></font></p>
    <div align="center">
    <table width="440" border="1" cellpadding="0" cellspacing="0">
    <tr bordercolor="#002E5B" bgcolor="#DCE0F1">

```

```

<td width="105"><div align="center"><strong><font color="#002E5B" size="2" face="Verdana, Arial,
elvetica, sans-serif">CODIGO</font></strong></div></td>
<td width="329" bgcolor="#DCE0F1"><div align="center"><strong><font color="#002E5B" size="2"
face="Verdana, Arial, Helvetica, sans-serif">DESCRIPCION</font></strong></div></td>
</tr>
<?php
    include 'Conexion.php';
    //Para Insertar Registros
    $valor1=$_HTTP_GET_VARS['valor'];
    if($valor1==1)
    {
        $NombreOrg = $_POST[txt_NombreOrg];
        $DireccionOrg = $_POST[txt_DireccionOrg];
        $TelefonoOrg = $_POST[txt_TelefonoOrg];
        $v_codigoorg = pg_exec("select nextval('seq_cod_org')");
        $filaSec = pg_numrows($v_codigoorg);
        $v_codigoorg1 = pg_result($v_codigoorg,0,0);
        //echo "<h4>$valor</h4>";
        $Contador = 1;
        //Hace una consulta para ver si cliente existe
        $result=pg_exec("SELECT * FROM organizacion where codigo_organizacion =
        $v_codigoorg1");
        //Ve el número de filas que obtiene el select si es 0 el cliente no existe
        $filas=pg_numrows($result);
        if($filas == 0){
            //echo " cliente no existe";
            // Inserta los datos a la tabla
            $sql="INSERT INTO organizacion (codigo_organizacion, nombre_organizacion,
            direccion_organizacion, telefono_organizacion) VALUES
            ($v_codigoorg1,$NombreOrg,$DireccionOrg,$TelefonoOrg)";
            pg_exec($conexion, $sql);
            // Muestra los datos de la tabla luego del ingreso del registro
            $result = pg_exec("SELECT codigo_organizacion, nombre_organizacion FROM
            organizacion order by codigo_organizacion");
            $filas = pg_num_rows($result);
            for($i=0;$i<$filas;$i++){
                $dato1 = pg_result($result,$i,0);
                $dato2 = pg_result($result,$i,1);
                if(ceil($Contador/2) == ($Contador/2)){
                    echo "<tr bgcolor='#FEF3FE' class='TextoTabla'>
                    <td width='95'><div align='left'><a href='Manten_Organizacion.php?CedCli=$dato1'>
                    $dato1</a></div></td>
                    <td width='220'><div align='left'>$dato2</div></td>
                    </tr>";
                }else{
                    echo "<tr bgcolor='#FFFFFF' class='TextoTabla'>
                    <td width='95'><div align='left'><a href='Manten_Organizacion.php?CedCli=$dato1'>
                    $dato1</a></div></td>
                    <td width='220'><div align='left'>$dato2</div></td>
                    </tr>";
                }
            }
            $Contador++;
        }
        }else{
            echo "<p align='center' class='TextoTabla1'>Organizacion ya existe: Datos no se
            grabaron</p>";
        }
    }
    //Para Modificar registros
    if($valor1==2)
    {
        $CodigoOrg = $_POST[txt_CodigoOrg];
        $NombreOrg = $_POST[txt_NombreOrg];
        $DireccionOrg = $_POST[txt_DireccionOrg];
    }

```



```

        <input name="btn_Volver" type="submit" id="btn_Volver" value="Volver">
    </div>
    <div align="center"></div></td></tr>
</table>
</div>
</form>
</TD> </TR> </TBODY></TABLE>
</body>
</html>

```

### Manten\_Organizacion.php

```

<html>
<head>
<title>Organizaciones - SISCON</title>
</head>
<body>
<?php
    $CodigoOrg1 = $_HTTP_GET_VARS['CedCli'];
    $Contador = 1;
    include 'Conexion.php';
    //Hace una consulta para ver datos de la habitacion
    $result=pg_exec("SELECT * FROM organizacion where codigo_organizacion = $CodigoOrg1");
    $filas=pg_numrows($result);
?>
<TABLE height=204 cellSpacing=0 cellPadding=0 width=725 align=left border=0>
<TBODY>
<TR>
<TD width=725 height=204><form action="Ingreso_Organizacion.php?valor=2"
    method="post"name="frm_UsuarioFinal2" id="frm_UsuarioFinal2">
    <p align="center"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif"><strong>ORGANIZACIONES</strong></font></p>
    <div align="center">
        <table width="636" border="0" cellpadding="0" cellspacing="0">
            <tr>
                <td width="160"><div align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">CODIGO:</font></div></td>
                <td width="476"><div align="left"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif"><?php echo "$CodigoOrg1"; ?>
                    <?php $CodigoOrg1= pg_result($result, 0, 0);
                    echo "<input name='txt_CodigoOrg' type='hidden' id='txt_CodigoOrg' value='$CodigoOrg1'>";
                </font></div></td></tr>
            <tr>
                <td><p align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">DESCRIPCION:</font></p></td>
                <td align="left">
                    <?php $NombreOrg1 = pg_result($result, 0, 1);
                    echo "<input name='txt_NombreOrg' type='text' id='txt_NombreOrg' size='65' maxlength='60' value='$NombreOrg1'>";
                <?>
            </div></td></tr>
        </table>
    <?php
        pg_FreeResult($result);
    ?>
    <table width="260" border="0" cellpadding="0" cellspacing="0">
        <tr>
            <td width="260">&nbsp;</td></tr>
        <tr>
            <td><div align="center">
                <input name="btn_Grabar" type="submit" id="btn_Grabar" value="Guardar Cambios">
            </div></td></tr>
        </table></div>

```

```

        <p align="center"><font size="2" face="Verdana, Arial, Helvetica, sans-serif"><a
        href="Ingreso_Organizacion.php?valor=3">Ver Organizaciones</a></font></p>
    </form></TD>
</TR> </TBODY></TABLE>
</body>
</html>

```

### Ingreso\_TipoUsuario.php

```

<html>
<head>
<title>Tipo de Usuario - SISCON</title>
</head>
<body>
<TABLE height=284 cellSpacing=0 cellPadding=0 width=592 align=center border=0>
  <TBODY>
    <TR>
      <TD vAlign=top width=91 >&nbsp;  </TD>
      <TD vAlign=top width=501><TABLE height=64 cellSpacing=0 cellPadding=0 width=488 align=center
        border=0>
        <TBODY>
          <TR>
            <TD width=468 height=64><form action="TipoUsuario.html" method="post" name="frm_Cliente1"
            id="frm_Cliente1">
              <p align="center"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
              serif"><strong>TIPO DE USUARIO </strong></font></p>
              <div align="center">
                <table width="440" border="1" cellpadding="0" cellspacing="0">
                  <tr bordercolor="#002E5B" bgcolor="#DCE0F1">
                    <td width="105"><div align="center"><strong><font color="#002E5B" size="2" face="Verdana,
                    Arial, Helvetica, sans-serif">CODIGO</font></strong></div></td>
                    <td width="329" bgcolor="#DCE0F1"><div align="center"><strong><font color="#002E5B"
                    size="2" face="Verdana, Arial, Helvetica, sans-serif">DESCRIPCION</font></strong></div></td>
                  </tr>
                </table>
                <?php
                  include 'Conexion.php';
                  //Para Insertar Registros
                  $valor1=$_HTTP_GET_VARS['valor'];
                  if($valor1==1)
                  {
                    $DescripcionTUsu = $_POST[txt_DescripcionTUsu];
                    v_codigotus = pg_exec("select nextval('seq_cod_tusu')");
                    $filaSec = pg_numrows($v_codigotus);
                    $v_codigotus1 = pg_result($v_codigotus,0,0);
                    $Contador = 1;
                    //Hace una consulta para ver si cliente existe
                    $result=pg_exec("SELECT * FROM tipo_usuario where codigo_usuario =
                    $v_codigotus1");
                    //Ve el n&uacute;mero de filas que obtiene el select si es 0 el cliente no existe
                    $filas=pg_numrows($result);
                    if ($filas==0)
                    {
                      // Inserta los datos a la tabla
                      $sql ="INSERT INTO tipo_usuario (codigo_usuario, descripcion_usuario) VALUES
                      ($v_codigotus1,$DescripcionTUsu)";
                      pg_exec($conexion, $sql);
                      // Muestra los datos de la tabla luego del ingreso del registro
                      $result = pg_exec("SELECT codigo_usuario, descripcion_usuario FROM tipo_usuario
                      order by codigo_usuario");
                      $filas = pg_num_rows($result);
                      for($i=0;$i<$filas;$i++){
                        $dato1 = pg_result($result,$i,0);
                        $dato2 = pg_result($result,$i,1);
                        if(ceil($Contador/2) == ($Contador/2))
                          echo "<tr bgcolor='#FEF3FE' class='TextoTabla'>

```

```

<td width='95'><div align='left'><a href='Manten_TipoUsuario.php?CedCli=$dato1'>
$dato1</a></div></td>
<td width='220'><div align='left'>$dato2</div></td> </tr>";
}
else
{
echo "<tr bgcolor='#FFFFFF' class='TextoTabla'>
<td width='95'><div align='left'><a href='Manten_TipoUsuario.php?CedCli=$dato1'>
$dato1</a></div></td>
<td width='220'><div align='left'>$dato2</div></td>
</tr>";
}
$Contador++;
} }
else
{
echo "<p align='center' class='TextoTabla1'>Tipo Usuario ya existe: Datos no se
grabaron</p>";
}}
//Para Modificar registros
if($valor1==2)
{
$CodigoTUsu = $_POST[txt_CodigoTUsu];
$DescripcionTUsu = $_POST[txt_DescripcionTUsu];
echo "<h4>$Nombres</h4>";
$Contador = 1;
// Modifica los datos a la tabla
$sql="UPDATE tipo_usuario SET descripcion_tusuario = '$DescripcionTUsu' WHERE
codigo_tusuario = $CodigoTUsu";
pg_exec($conexion, $sql);
// Muestra los datos de la tabla luego del ingreso del registro
$result = pg_exec("SELECT codigo_tusuario, descripcion_tusuario FROM tipo_usuario
order by codigo_tusuario");
$filas = pg_num_rows($result);
for($i=0;$i<$filas;$i++){
$dato1 = pg_result($result,$i,0);
$dato2 = pg_result($result,$i,1);
if(ceil($Contador/2) == ($Contador/2))
{
echo "<tr bgcolor='#FEF3FE' class='TextoTabla'>
<td width='95'><div align='left'><a href='Manten_TipoUsuario.php?CedCli=$dato1'>
$dato1</a></div></td>
<td width='220'><div align='left'>$dato2</div></td>
</tr>";
}
else
{
echo "<tr bgcolor='#FFFFFF' class='TextoTabla'>
<td width='95'><div align='left'><a href='Manten_TipoUsuario.php?CedCli=$dato1'>
$dato1</a></div></td>
<td width='220'><div align='left'>$dato2</div></td></tr>";
}
$Contador++;
} }
//Para ver registros
if($valor1==3)
{
$DescripcionTUsu = $_POST[txt_DescripcionTUsu];
$Contador = 1;
// Muestra los datos de la tabla luego del ingreso del registro
$result = pg_exec("SELECT codigo_tusuario, descripcion_tusuario FROM tipo_usuario
order by codigo_tusuario");
$filas = pg_num_rows($result);

```



[illegible]

```

<TR>
<TD width=468 height=64><form action="Ingreso_TipoUsuario.php?valor=2" method="post"
name="frm_UsuarioFinal2" id="frm_UsuarioFinal2">
<p align="center"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
serif"><strong>TIPO USUARIO</strong></font></p>
<div align="center">
<table width="424" border="0" cellpadding="0" cellspacing="0">
<tr>
<td width="112"><div align="left"><font color="#002E5B" size="2" face="Verdana, Arial,
Helvetica, sans-serif">CODIGO:</font></div></td>
<td width="312"><div align="left"><font color="#002E5B" size="2" face="Verdana, Arial,
Helvetica, sans-serif"><?php echo "$CodigoTUsu1"; ?>
<?php $CodigoTUsu1= pg_result($result, 0, 0);
echo "<input name='txt_CodigoTUsu' type='hidden' id='txt_CodigoTUsu'
value='$CodigoTUsu1'>";
?>
</font></div></td></tr>
<tr>
<td><p align="left"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
serif">DESCRIPCION:</font></p></td>
<td><div align="left">
<?php $DescripcionTUsu1= pg_result($result, 0, 1);
echo "<input name='txt_DescripcionTUsu' type='text' id='txt_DescripcionTUsu'
size='60' maxlength='50' value='$DescripcionTUsu1'>";
?>
</div></td></tr>
</table>
<?php
pg_FreeResult($result);
?>
<table width="260" border="0" cellpadding="0" cellspacing="0">
<tr>
<td width="260">&nbsp;</td>
</tr>
<tr>
<td><div align="center">
<input name="btn_Grabar" type="submit" id="btn_Grabar" value="Guardar Cambios">
</div></td></tr>
</table>
<p><font size="2" face="Verdana, Arial, Helvetica, sans-serif"><a
href="Ingreso_TipoUsuario.php?valor=3">Ver Tipos de Usuario</a></font></p>
</div>
<p align="center">&nbsp;</p>
</form>
<div align="justify"></div></TD>
</TR>
</TBODY>
</TABLE>
<DIV align=center></DIV></TD>
</TR> </TBODY></TABLE>
</body>
</html>

```

#### Ingresos.php

```

<?php
session_start();
if (isset($_SESSION['k_username'])) {
    $usuario1 = $_SESSION['k_username'];
    //echo "zdiego.$usuario1";
}else{
    echo "<TABLE cellSpacing=0 cellPadding=0 width=718 border=0 align=center>
<TBODY>
<TR>
<TD><IMG height=62

```

```

src='Graficos/logo.gif' width=177></TD>
<TD width=541 bgColor=#9a4d9f><div align='center'><font face = 'Verdana' color = 'white' size = '3'>
strong>DEBE INICIAR LA SESION CON UNA CUENTA EXISTENTE</strong></font></div></TD>
</TR>
</TABLE>
<div align='center'><font face = 'Verdana' color = 'white' size = '2'><strong><a
ref='Index.html'>REGRESAR A PRINCIPAL</a></strong></font></div>";
exit();
}
?>
<html>
<head>
<title>Egresos - SISCON</title>
</head>
<body>
<?php
include 'Conexion.php';
$Fecha1 = getdate();
$FechaE = $Fecha1[year]."-".$Fecha1[mon]."-".$Fecha1[mday];
$Anio = $Fecha1[year];
?>
<TABLE width=592 height=284 border=0 align=center cellPadding=0 cellSpacing=0>
<TBODY>
<TR>
<TD vAlign=top width=501><TABLE height=64 cellSpacing=0 cellPadding=0 width=576 align=center
border=0>
<TBODY>
<TR>
<TD width=576 height=64><form action="Envio_Ingresos.php" method="post"
name="frm_UsuarioFinal" id="frm_UsuarioFinal">
<p align="center"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
serif"><strong>CABECERA</strong></font></p>
<div align="right">
<table width="585" border="0" cellpadding="0" cellspacing="0">
<tr>
<td width="175"><div align="right"><font color="#002E5B" size="2" face="Verdana, Arial,
Helvetica, sans-serif">A&Ntilde;O TRANSACCION:</font></div></td>
<td width="401">
<div align="left"><strong><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica,
sans-serif">
<?php echo "<input name='txt_Anio' type='hidden' id='txt_Anio' value='$Anio'>$Anio"; ?>
</font></strong> </div></td></tr>
<tr>
<td><p align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
serif">PROVEEDOR:</font></p></td>
<td>
<div align="left">
<?php
$result = pg_exec("SELECT codigo_proveedor, ruc, razonsocial_proveedor
FROM proveedor
order by razonsocial_proveedor");
$filas = pg_num_rows($result);
echo "<select name='cmb_Proveedor'>";
for($i=0;$i<$filas;$i++)
{
$dato1 = pg_result($result,$i,0);
$dato2 = pg_result($result,$i,1);
$dato3 = pg_result($result,$i,2);
echo "<option value='$dato1'>$dato2: $dato3</option>";
}
echo "</select>";
?>
</div></td></tr>
<tr>

```

```

<td><div align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">ORGANIZACION:</font></div></td>
<td>
<div align="left">
<?php
$result = pg_exec("SELECT codigo_organizacion, nombre_organizacion
FROM organizacion
order by codigo_organizacion");
$filas = pg_num_rows($result);
echo "<select name='cmb_Organizacion'>";
for($i=0;$i<$filas;$i++)
{
$dato1 = pg_result($result,$i,0);
$dato2 = pg_result($result,$i,1);
echo "<option value='$dato1'>$dato2</option>";
}
echo "</select>";
?>
</div></td></tr>
<tr>
<td><div align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">FECHA TRANSACCION:</font></div></td>
<td>
<div align="left"><strong><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">
<?php echo "<input name='txt_FechaTI' type='hidden' id='txt_FechaTI' value='$FechaE'>$FechaE";
?>
</font></strong> </div></td>
</tr>
<tr>
<td><div align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">DETALLE DE TRANSACCION:</font></div></td>
<td>
<div align="left">
<textarea name="txt_DetalleTI" cols="40" rows="4" id="txt_DetalleTI"></textarea>
</div></td></tr>
<tr>
<td><div align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">TIPO DE DOCUMENTO:</font></div></td>
<td><div align="left">
<?php
$result = pg_exec("SELECT codigo_tdocumento, descripcion_tdocumento
FROM tipodocumento
order by codigo_tdocumento");
$filas = pg_num_rows($result);
echo "<select name='cmb_TDocumento'>";
for($i=0;$i<$filas;$i++)
{
$dato1 = pg_result($result,$i,0);
$dato2 = pg_result($result,$i,1);
echo "<option value='$dato1'>$dato2</option>";
}
echo "</select>";
?>
</div></td></tr>
<tr>
<td><div align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">TIPO DE TRANSACCION:</font></div></td>
<td><div align="left">
<?php
$result1 = pg_exec("SELECT codigo_ttransaccion, descripcion_ttransaccion
FROM objetotransaccion
order by codigo_ttransaccion");
$filas1 = pg_num_rows($result1);

```

```

        echo "<select name='cmb_TTransaccion'>";
        for($i=0;$i<$filas1;$i++)
        {
            $dato1_1 = pg_result($result1,$i,0);
            $dato2_1 = pg_result($result1,$i,1);
            "<option value='$dato1_1'>$dato2_1</option>";
        }
        echo "</select>";
    ?>
</div></td></tr>
</table>
</div>
<table width="260" border="0" align="center" cellpadding="0" cellspacing="0">
<tr>
<td width="260"><strong><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica,
    sans-serif">
<?php echo "<input name='txt_CodigoF' type='hidden' id='txt_CodigoF' value='$usuario1'>"; ?>
</font></strong></td>
<td width="260">&nbsp;   </td></tr>
<tr>
<td><div align="center">
<input name="btn_Grabar" type="submit" id="btn_Grabar" value="Guardar">
</div></td>
<td><div align="center">
<input name="btn_Cancelar" type="reset" id="btn_Cancelar" value="Cancelar">
</div></td></tr>
</table>
<p align="center"><font size="2" face="Verdana, Arial, Helvetica, sans-serif"><a
href="ver_ingresos.php"
>Ver Ingresos </a></font></p>
</form>
</TD>
</TR> </TBODY> </TABLE>
<DIV align=center></DIV></TD>
</TR> </TBODY></TABLE>
</body>
</html>

Ver_ingresos.php
<html>
<head>
<title>EGRESOS - SISCON</title>
</head>
<body>
<div align="right"></div>
<?php
    include 'Conexion.php';
?>
<table width="681" border="0" align="center" cellpadding="0" cellspacing="0" bgcolor="#FFFFFF">
<tr>
<td width="681" height="152"><br>
<table width="655" border="0" align="center" cellpadding="0" cellspacing="0">
<tr>
<td width="645" align="center" valign="top"><p align="center"><font color="#002E5B" size="2"
    face="Verdana, Arial, Helvetica, sans-serif"><strong>INGRESOS</strong></font></p>
<div align="center">
<table width="650" border="1" cellpadding="0" cellspacing="0">
<tr bordercolor="#002E5B" bgcolor="#DCE0F1">
<td width="68"><div align="center"><strong><font color="#002E5B" size="2" face="Verdana,
    Arial, Helvetica, sans-serif">CODIGO</font></strong></div></td>
<td width="232" bgcolor="#DCE0F1"><div align="center"><strong><font color="#002E5B"
    size="2" face="Verdana, Arial, Helvetica, sans-serif">PROVEEDOR</font></strong></div></td>
<td width="227" bgcolor="#DCE0F1"><div align="center"><strong><font color="#002E5B"
    size="2" face="Verdana, Arial, Helvetica, sans-serif">ORGANIZACION</font></strong></div></td>

```

```

        <td width="113" bgcolor="#DCE0F1"><div align="center"><strong><font color="#002E5B"
size="2" face="Verdana, Arial, Helvetica, sans-serif">FECHA TRANS. </font></strong></div></td>
    </tr>
<?php
    //Para ver registros
    $Contador = 1;
    $result = pg_exec(" SELECT I.codigo_ingreso, I.anio_ingreso, P.razonsocial_proveedor,
    O.nombre_organizacion, I.fecha_transaccioni
    FROM ingreso I, proveedor P, organizacion O
    WHERE          I.codigo_proveedor = P.codigo_proveedor
    AND            I.codigo_organizacion = O.codigo_organizacion");
    $filas = pg_num_rows($result);
    for($i=0;$i<$filas;$i++){
        $dato1 = pg_result($result,$i,0);
        $dato2 = pg_result($result,$i,1);
        $dato3 = pg_result($result,$i,2);
        $dato4 = pg_result($result,$i,3);
        $dato5 = pg_result($result,$i,4);
        if(ceil($Contador/2) == ($Contador/2)){
            echo "<tr bgcolor='#FEF3FE' class='TextoTabla'>
            <td><div align='left'><a href='Manten_Ingreso.php?CedCli=$dato1'>($dato2)-
            $dato1</a></div></td>
            <td><div align='left'>$dato3</div></td>
            <td><div align='left'>$dato4</div></td>
            <td><div align='left'>$dato5</div></td>
            <tr>";
        }else{
            echo "<tr bgcolor='#FFFFFF' class='TextoTabla'>
            <td><div align='left'><a href='Manten_Ingreso.php?CedCli=$dato1'>($dato2)-
            $dato1</a></div></td>
            <td><div align='left'>$dato3</div></td>
            <td><div align='left'>$dato4</div></td>
            <td><div align='left'>$dato5</div></td>
            </tr>";
        }
        $Contador++;
    }
    pg_FreeResult($result);

?>
</table>
</div></td></tr>
</table>
<br>
<form name="form1" method="post" action="Ingresos.php">
    <table width="200" border="0" align="center" cellpadding="0" cellspacing="0">
        <tr>
            <td><div align="center">
                <input type="submit" name="Submit" value="Volver">
            </div></td></tr>
        </table>
    </form></td></tr>
</table>
</body>
</html>

Envio_Ingresos.php
<html>
<head>
<title>Untitled Document</title>
</head>
<body>
<?php
    $Anio = $_POST[txt_Anio];
    $Proveedor = $_POST[cmb_Proveedor];
    $Organizacion = $_POST[cmb_Organizacion];

```

```

$FechaTI = $_POST[txt_FechaTI];
$DetalleTI = $_POST[txt_DetalleTI];
$CodigoF = $_POST[txt_CodigoF];
$TDocumento = $_POST[cmb_TDocumento];
$TTransaccion = $_POST[cmb_TTransaccion];
include 'Conexion.php';
$v_codigoingre = pg_exec("select nextval('seq_cod_ing')");
$filaSec = pg_numrows($v_codigoingre);
$v_codigoingre1 = pg_result($v_codigoingre,0,0);
// Inserta los datos a la tabla
$sql = "INSERT INTO ingreso (codigo_ingreso, anio_ingreso, codigo_proveedor,
codigo_organizacion, fecha_transaccioni, detalle_transaccioni, codigo_funcionaria,
codigo_tdocumento, codigo_ttransaccion)
VALUES ($v_codigoingre1,'$Anio', $Proveedor, $Organizacion, '$FechaTI', '$DetalleTI', $CodigoF,
$TDocumento, $TTransaccion)";
pg_exec($conexion, $sql);
//echo "$sql";
?>

<?php
echo "<SCRIPT LANGUAGE='javascript'>
location.href = 'Ingresos_Productosl.php?Ingr=$v_codigoingre1&AnioI=$Anio';
</SCRIPT>";

?>
</body>
</html>

```

### Ingresos\_Productosl.php

```

<?php
include 'Conexion.php';
$CodigoIn = $_HTTP_GET_VARS['Ingr'];
$AnioIn = $_HTTP_GET_VARS['AnioI'];
$result = pg_exec("SELECT I.codigo_ingreso, I.anio_ingreso, P.codigo_proveedor,
P.razonsocial_proveedor, O.codigo_organizacion, O.nombre_organizacion, I.fecha_transaccioni,
I.detalle_transaccioni, TD.codigo_tdocumento, TD.descripcion_tdocumento,
OT.codigo_ttransaccion, OT.descripcion_ttransaccion, F.codigo_funcionaria
FROM ingreso I, proveedor P, organizacion O, tipodocumento TD, objetotransaccion OT,
funcionaria F
WHERE I.codigo_proveedor = P.codigo_proveedor
AND I.codigo_organizacion = O.codigo_organizacion
AND I.codigo_tdocumento = TD.codigo_tdocumento
AND I.codigo_ttransaccion = OT.codigo_ttransaccion
AND I.codigo_funcionaria = F.codigo_funcionaria
AND I.codigo_ingreso = ".$CodigoIn." AND I.anio_ingreso = ".$AnioIn."");
$filas=pg_numrows($result);

?>
<html>
<head>
<title>Saving Partners - Proceso de Compras Web</title>
</head>
<body>
<div align="right"></div>
<table width="607" border="0" align="center" cellpadding="0" cellspacing="0" bgcolor="#FFFFFF">
<tr>
<td width="607" height="453"><div align="center">
<table width="555" border="0" align="center" cellpadding="0" cellspacing="0">
<tr>
<td><form name="form2" method="post" action="Ingresos_Productosl.php?Ingr=<?php echo
"$CodigoIn"; ?>&AnioI=<?php echo "$AnioIn"; ?>">
<table width="100%" border="0" align="center" cellpadding="0" cellspacing="0">
<tr>
<td height="23" colspan="7"><div align="center"><span class="Estilo23">CABECERA
</span></div></td></tr>
<tr bgcolor="#6CCFFF">

```

```

<td bgcolor="#FBEADB"><div align="right"><span class="Estilo23">Codigo
Transaccion:</span></div></td>
<td bgcolor="#FFFFFF">&nbsp;</td>
<td bgcolor="#FFFFFF"><div align="left"><span class="Estilo25"><font color="#002E5B"
size="2" face="Verdana, Arial, Helvetica, sans-serif">
<?php
    $Codigol1= pg_result($result, 0, 0);
    echo "$Codigol1";
    echo "<input name='txt_Codigol' type='hidden' id='txt_Codigol' value='$Codigol1'>";
?>
</font> </span></div></td>
<td bgcolor="#FFFFFF">&nbsp;</td>
<td bgcolor="#FFFFFF">&nbsp;</td>
<td bgcolor="#FFFFFF">&nbsp;</td>
<td bgcolor="#FFFFFF">&nbsp;</td></tr>
<tr bgcolor="#6CCFFF">
<td width="17%" bgcolor="#0091D7"><div align="right" class="Estilo23">A&ntilde;o
Transaccion:</div></td>
<td width="3%" bgcolor="#6CCFFF">&nbsp;</td>
<td width="33%" bgcolor="#6CCFFF"><font color="#002E5B" size="2" face="Verdana, Arial,
Helvetica, sans-serif">
<span class="Estilo25"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
serif">
<?php
    $Aniol1= pg_result($result, 0, 1);
    echo "$Aniol1";
    echo "<input name='txt_Anio' type='hidden' id='txt_Anio' value='$Aniol1'>";
?>
</font></span> </font></td>
<td width="3%" bgcolor="#FFFFFF">&nbsp;</td>
<td width="14%" bgcolor="#0091D7"><div align="right" class="Estilo23">Fecha:</div></td>
<td width="2%" bgcolor="#6CCFFF">&nbsp;</td>
<td width="28%" bgcolor="#6CCFFF"> <font color="#002E5B" size="2" face="Verdana, Arial,
Helvetica, sans-serif">
<span class="Estilo25"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
serif">
<font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">
<?php
    $FechaTI1= pg_result($result, 0, 6);
    echo "$FechaTI1";
    echo "<input name='txt_FechaTI' type='hidden' id='txt_FechaTI' value='$FechaTI1'>";
?>
</font> </font></span> </font></td></tr>
<tr>
<td bgcolor="#FBEADB"><div align="right" class="Estilo23">Proveedor:</div></td>
<td>&nbsp;</td>
<td><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">
<span class="Estilo25"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
serif">
<?php
    $CodigoP1= pg_result($result, 0, 2);
    $NombreP1= pg_result($result, 0, 3);
    "$NombreP1";
    echo "<input name='txt_Proveedor' type='hidden' id='txt_Proveedor' value='$CodigoP1'>";
?>
</font></span> </font></td>
<td bgcolor="#FFFFFF">&nbsp;</td>
<td bgcolor="#FBEADB"><div align="right" class="Estilo23">Tipo de Documento:</div>
<div align="right"></div></td>
<td>&nbsp;</td>
<td class="Estilo25"><div align="left"><font color="#002E5B" size="2" face="Verdana, Arial,
Helvetica, sans-serif">
<font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">
<?php

```



```

        $CodigoTD1= pg_result($result, 0, 8);
        $NombreTD1= pg_result($result, 0, 9);
        echo "$NombreTD1";
        echo "<input name='txt_TDocumento' type='hidden' id='txt_TDocumento'
        value='$CodigoTD1'>";
        ?>
    </font> </font></div></td></tr>
<tr bgcolor="#6CCFFF">
    <td bgcolor="#0091D7"><div align="right"
class="Estilo25"><strong>Organizacion:</strong></div></td>
    <td bgcolor="#6CCFFF">&nbsp;</td>
    <td bgcolor="#6CCFFF"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica,
    sans-serif">
    <span class="Estilo25"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
    serif">
    <?php
        $CodigoO1= pg_result($result, 0, 4);
        $NombreO1= pg_result($result, 0, 5);
        echo "$NombreP1";
        echo "<input name='txt_Organizacion' type='hidden' id='txt_Organizacion'
        value='$CodigoO1'>";
    ?>
    </font></span> </font></td>
    <td bgcolor="#FFFFFF">&nbsp;</td>
    <td bgcolor="#6CCFFF"><div align="right"><span class="Estilo25"><span
    class="Estilo23">Tipo de Transaccion:</span></span></div></td>
    <td bgcolor="#6CCFFF">&nbsp;</td>
    <td bgcolor="#6CCFFF"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
    serif">
    <span class="Estilo25"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
    serif">
    <font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">
    <?php
        $CodigoOT1= pg_result($result, 0, 10);
        $NombreOT1= pg_result($result, 0, 11);
        echo "$NombreOT1";
        echo "<input name='txt_TTransaccion' type='hidden' id='txt_TTransaccion'
        value='$CodigoOT1'>";
    ?>
    </font> </font></span> </font></td></tr>
<tr bgcolor="#6CCFFF">
    <td bgcolor="#0091D7"><div align="right"><span class="Estilo25"><strong> Detalle:
    </strong></span></div></td>
    <td bgcolor="#6CCFFF">&nbsp;</td>
    <td bgcolor="#6CCFFF"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
    serif">
    <span class="Estilo25"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
    serif"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">
    <?php
        $DetalleT11= pg_result($result, 0, 7);
        $CodigoF1= pg_result($result, 0, 12);
        echo "$DetalleT11";
        echo "<input name='txt_DetalleTI' type='hidden' id='txt_DetalleTI' value='$DetalleT11'>";
        echo "<input name='txt_CodigoF' type='hidden' id='txt_CodigoF' value='$CodigoF1'>";
    ?>
    </font></font></span> </font></td>
    <td bgcolor="#FFFFFF">&nbsp;</td>
    <td bgcolor="#6CCFFF">&nbsp;</td>
    <td bgcolor="#6CCFFF">&nbsp;</td>
    <td bgcolor="#6CCFFF">&nbsp;</td></tr>
<tr>
    <td height="52" colspan="7"><div align="center">
    <hr>
    <?php

```

```

$CodigoIn1 = $HTTP_GET_VARS['Ingr'];
$AnioIn1 = $HTTP_GET_VARS['Anio1'];
$Producto = $_POST[cmb_Producto];
$Cantidad = $_POST[txt_Cantidad];
$Contador = 1;
$v_codigoingr = pg_exec("select nextval('seq_cod_dingr')");
$filaSec = pg_numrows($v_codigoingr);
$v_codigodogr1 = pg_result($v_codigoingr,0,0);
if($Producto != ""){
$sql="INSERT INTO detalle_ingreso (codigo_dingreso, codigo_ingreso, anio_ingreso,
codigo_producto, cantidad) VALUES ($v_codigodogr1, $CodigoIn1, $AnioIn1,
'$Producto', $Cantidad)";
pg_exec($conexion, $sql);
}
?>
<table width="549" border="0" cellpadding="0" cellspacing="0">
<tr>
<td height="26" colspan="7"> <div align="right" class="Estilo25">
<div align="center"><strong>PRODUCTOS</strong></div>
</div></td></tr>
<tr class="Estilo25">
<td width="87" class="Estilo25"><div align="right" class="Estilo25">
<div align="right"><strong>Producto:</strong></div>
</div></td>
<td width="9"><div align="left">
</div></td>
<td width="250"><div align="left">
<?php
    $resultP1 = pg_exec("SELECT codigo_producto, descripcion_producto
    FROM producto
    order by descripcion_producto");
    $filasP1 = pg_num_rows($resultP1);
    echo "<select name='cmb_Producto'>";
    for($i=0;$i<$filasP1;$i++)
    {
        $dato1_1 = pg_result($resultP1,$i,0);
        $dato2_1 = pg_result($resultP1,$i,1);
        echo "<option value='$dato1_1'>$dato2_1</option>";
    }
    echo "</select>";
?>
</div></td>
<td width="10"><div align="right"></div></td>
<td width="109"><div align="right"></div></td>
<td width="7">&nbsp;</td>
<td width="77"><div align="left"></div></td></tr>
<tr class="Estilo25">
<td width="87"><div align="right"><strong>Descripci&ocaron;:</strong></div></td>
<td><div align="left"></div></td>
<td><div align="left">
<input name="txt_DescripcionProd" type="text" id="txt_DescripcionProd" size="40"
maxlength="50">
</div></td>
<td><div align="right"></div></td>
<td><div align="right"></div></td>
<td>&nbsp;</td>
<td><div align="left"></div></td></tr>
<tr class="Estilo25">
<td><div align="right"><strong>Cantidad:</strong></div></td>
<td>&nbsp;</td>
<td><div align="left">
<input name="txt_Cantidad" type="text" id="txt_CodigoProd3" size="10" maxlength="5">
</div></td>
<td>&nbsp;</td>

```

```

        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td></tr>
    <tr class="Estilo25">
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
    </tr>
    <tr class="Estilo25">
        <td colspan="7"><div align="center">
            <input name="btn_Guardar" type="submit" id="btn_Guardar" value="A&ntilde;adir
            Productos">
        </div></td></tr></table>
    <hr>
    </div></td></tr></table>
</form></td></tr></table>
</div>
<form name="form1" method="post" action="Ingresos.php">
<div align="center">
<table width="563" border="1" cellpadding="0" cellspacing="0">
<tr bordercolor="#002E5B" bgcolor="#DCE0F1">
<td width="75"><div align="center"><strong><font color="#002E5B" size="2" face="Verdana, Arial,
Helvetica, sans-serif">CODIGO</font></strong></div></td>
<td width="383" bgcolor="#DCE0F1"><div align="center"><strong><font color="#002E5B" size="2"
face="Verdana, Arial, Helvetica, sans-serif">DESCRIPCION</font></strong></div></td>
<td width="97" bgcolor="#DCE0F1"><div align="center"><strong><font color="#002E5B" size="2"
face="Verdana, Arial, Helvetica, sans-serif">CANTIDAD</font></strong></div></td></tr>
<?php
    $Contador = 1;
    Hace una consulta para ver si cliente existe
    $resultIn = pg_exec("SELECT I.codigo_ingreso, I.anio_ingreso, DI.codigo_producto,
    P.descripcion_producto, DI.cantidad
    FROM detalle_ingreso DI, ingreso I, producto P
    WHERE DI.codigo_ingreso = I.codigo_ingreso
    AND DI.anio_ingreso = I.anio_ingreso
    AND DI.codigo_producto = P.codigo_producto
    AND DI.codigo_ingreso = $CodigoIn
    AND DI.anio_ingreso = $AnioIn");
    $filasIn = pg_num_rows($resultIn);
    for($i=0;$i<$filasIn;$i++){
        $datol1 = pg_result($resultIn,$i,2);
        $datol2 = pg_result($resultIn,$i,3);
        $datol3 = pg_result($resultIn,$i,4);
        if(ceil($Contador/2) == ($Contador/2)){
            echo "<tr bgcolor='#FEF3FE' class='TextoTabla'>
            <td><div align='left'><a href='Manten_DIngreso.php?CedCli=$datol1'>
            $datol1</a></div></td>
            <td><div align='left'>$datol2</div></td>
            <td><div align='left'>$datol3</div></td></tr>";
        }else{
            echo "<tr bgcolor='#FFFFFF' class='TextoTabla'>
            <td><div align='left'><a href='Manten_DIngreso.php?CedCli=$datol1'>
            $datol1</a></div></td>
            <td><div align='left'>$datol2</div></td>
            <td><div align='left'>$datol3</div></td>
            </tr>";
        }
        $Contador++;
    }
    pg_FreeResult($result);
?>
</table>

```

```

        <br>
        <input type="submit" name="Submit" value="Finalizar">
    </div>
</form>
    <p align="center">&nbsp;</p></td></tr>
</table>
</body>
</html>

Egresos.php
<?php
session_start();
if (isset($_SESSION['k_username'])) {
    $usuario1 = $_SESSION['k_username'];
}else{
    echo "<TABLE cellSpacing=0 cellPadding=0 width=718 border=0 align=center>
    <TBODY>
    <TR>
    <TD><IMG height=62
    src='Graficos/logo.gif' width=177></TD>
    <TD width=541 bgColor=#9a4d9f><div align='center'><font face = 'Verdana' color = 'white' size = '3'>
    <strong>DEBE INICIAR LA SESION CON UNA CUENTA EXISTENTE</strong>
    </font></div></TD>
    </TR>
    </TABLE>
    <div align='center'><font face = 'Verdana' color = 'white' size = '2'><strong><a
    href='Index.html'>REGRESAR A PRINCIPAL</a></strong></font></div>";
    exit();
}
?>
<html>
<head>
<title>Egresos - SISCON</title>
</head>
<body>
<?php
include 'Conexion.php';
$Fecha1 = getdate();
$FechaE = $Fecha1[year]."-".$Fecha1[mon]."-".$Fecha1[mday];
$Anio = $Fecha1[year];
?>
<TABLE width=592 height=284 border=0 align=center cellPadding=0 cellSpacing=0>
<TBODY>
<TR>
<TD vAlign=top width=501><TABLE height=64 cellSpacing=0 cellPadding=0 width=576 align=center
border=0>
<TBODY>
<TR>
<TD width=576 height=64><form action="Envio_Egresos.php" method="post"
name="frm_UsuarioFinal" id="frm_UsuarioFinal">
<p align="center"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
serif"><strong>CABECERA</strong></font></p>
<div align="right">
<table width="585" border="0" cellpadding="0" cellspacing="0">
<tr>
<td width="175"><div align="right"><font color="#002E5B" size="2" face="Verdana, Arial,
Helvetica, sans-serif">A&Ntilde;O TRANSACCION:</font></div></td>
<td width="401">
<div align="left"><strong><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica,
sans-serif">
<?php echo "<input name='txt_Anio' type='hidden' id='txt_Anio' value='$Anio'>$Anio"; ?>
</font></strong> </div></td></tr>
</table>

```

```
 <p align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">RESPONSABLE:</font></p></td> <td> <div align="left"> <?php     $resultRes = pg_exec("SELECT codigo_responsable, cedula_responsable,     apellido_responsable, nombre_responsable     FROM responsable     order by apellido_responsable, nombre_responsable");     $filasRes = pg_num_rows($resultRes);     echo "<select name='cmb_Responsable'>";     for($i=0;$i<$filasRes;$i++)     {         $dato1 = pg_result($resultRes,$i,0);         $dato2 = pg_result($resultRes,$i,1);         $dato3 = pg_result($resultRes,$i,2);         $dato4 = pg_result($resultRes,$i,3);         echo "<option value='$dato1'>$dato2: $dato3 $dato4</option>";     }     echo "</select>"; ?> </div></td></tr> <tr> <td> <div align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">USUARIO FINAL:</font></div></td> <td> <div align="left"> <?php     $resultUF = pg_exec("SELECT codigo_ufinal, descripcion_ufinal     FROM usuario_final     order by codigo_ufinal");     $filasUF = pg_num_rows($resultUF);     echo "<select name='cmb_UFinal'>";     for($i=0;$i<$filasUF;$i++)     {         $dato1 = pg_result($resultUF,$i,0);         $dato2 = pg_result($resultUF,$i,1);         echo "<option value='$dato1'>$dato2</option>";     }     echo "</select>"; ?> </div></td></tr> <tr> <td> <div align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">UBICACION GEOGRAFICA:</font></div></td> <td> <div align="left"> <?php     $resultUb = pg_exec("SELECT codigo_ubicacion, descripcion_ubicacion     FROM destinogeo     order by codigo_ubicacion");     $filasUb = pg_num_rows($resultUb);     echo "<select name='cmb_Ubicacion'>";     for($i=0;$i<$filasUb;$i++)     {         $dato1 = pg_result($resultUb,$i,0);         $dato2 = pg_result($resultUb,$i,1);         echo "<option value='$dato1'>$dato2</option>";     }     echo "</select>"; ?> </div></td> </tr> <tr> |
```

```

<td><div align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">ORGANIZACION:</font></div></td>
<td>
<div align="left">
<?php
    $resultO = pg_exec("SELECT codigo_organizacion, nombre_organizacion
    FROM organizacion
    order by codigo_organizacion");
    $filasO = pg_num_rows($resultO);
    echo "<select name='cmb_Organizacion'>";
    for($i=0;$i<$filasO;$i++)
    {
        $dato1 = pg_result($resultO,$i,0);
        $dato2 = pg_result($resultO,$i,1);
        echo "<option value='$dato1'>$dato2</option>";
    }
    echo "</select>";
?>
</div></td></tr>
<tr>
<td><div align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">FECHA TRANSACCION:</font></div></td>
<td>
<div align="left"><strong><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">
<?php echo "<input name='txt_FechaT' type='hidden' id='txt_FechaT' value='$FechaE'>$FechaE";
?> </font></strong> </div></td>
</tr>
<tr>
<td><div align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">DETALLE DE TRANSACCION:</font></div></td>
<td>
<div align="left">
<textarea name="txt_DetalleT" cols="40" rows="4" id="txt_DetalleT"></textarea>
</div></td></tr>
<tr>
<td><div align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">TIPO DE DOCUMENTO:</font></div></td>
<td><div align="left">
<?php
    $result = pg_exec("SELECT codigo_tdocumento, descripcion_tdocumento
    FROM tipodocumento
    order by codigo_tdocumento");
    $filas = pg_num_rows($result);
    echo "<select name='cmb_TDocumento'>";
    for($i=0;$i<$filas;$i++)
    {
        $dato1 = pg_result($result,$i,0);
        $dato2 = pg_result($result,$i,1);
        echo "<option value='$dato1'>$dato2</option>";
    }
    echo "</select>";
?>
</div></td></tr>
<tr>
<td><div align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">TIPO DE TRANSACCION:</font></div></td>
<td><div align="left">
<?php
    $result1 = pg_exec("SELECT codigo_ttransaccion, descripcion_ttransaccion
    FROM objetotransaccion
    order by codigo_ttransaccion");
    $filas1 = pg_num_rows($result1);
    echo "<select name='cmb_TTransaccion'>";

```

```

        for($i=0;$i<$filas1;$i++)
        {
            $dato1_1 = pg_result($result1,$i,0);
            $dato2_1 = pg_result($result1,$i,1);
            echo "<option value='$dato1_1'>$dato2_1</option>";
        }
        echo "</select>";
    ?>
</div></td></tr>
</table>
</div>
<table width="260" border="0" align="center" cellpadding="0" cellspacing="0">
    <tr>
        <td width="260"><strong><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica,
            sans-serif">
<?php echo "<input name='txt_CodigoF' type='hidden' id='txt_CodigoF' value='$usuario1'>"; ?>
</font></strong></td>
        <td width="260">&nbsp;</td></tr>
        <tr>
            <td><div align="center">
                <input name="btn_Grabar" type="submit" id="btn_Grabar" value="Guardar">
            </div></td>
            <td><div align="center">
                <input name="btn_Cancelar" type="reset" id="btn_Cancelar" value="Cancelar">
            </div></td></tr></table>
        <p align="center"><font size="2" face="Verdana, Arial, Helvetica, sans-serif"><a
            href="ver_egresos.php">Ver Egresos </a></font></p>
    </form>
</TD>
</TR>    </TBODY>    </TABLE>
<DIV align=center></DIV></TD>
</TR> </TBODY></TABLE>
</body>
</html>

```

**Ver\_egresos.php**

```

<html>
<head>
<title>EGRESOS - SISCON</title>
</head>
<body>
<div align="right"></div>
<?php
    include 'Conexion.php';
?>
<table width="681" border="0" align="center" cellpadding="0" cellspacing="0" bgcolor="#FFFFFF">
    <tr>
        <td width="681" height="152"><br>
            <table width="655" border="0" align="center" cellpadding="0" cellspacing="0">
                <tr>
                    <td width="645" align="center" valign="top"><p align="center"><font color="#002E5B" size="2"
                        face="Verdana, Arial, Helvetica, sans-serif"><strong>EGRESOS</strong></font></p>
                    <div align="center">
                        <table width="650" border="1" cellpadding="0" cellspacing="0">
                            <tr bordercolor="#002E5B" bgcolor="#DCE0F1">
                                <td width="68"><div align="center"><strong><font color="#002E5B" size="2" face="Verdana,
                                    Arial, Helvetica, sans-serif">CODIGO</font></strong></div></td>
                                <td width="232" bgcolor="#DCE0F1"><div align="center"><strong><font color="#002E5B"
                                    size="2" face="Verdana, Arial, Helvetica, sans-serif">USUARIO FINAL
                                </font></strong></div></td>
                                <td width="227" bgcolor="#DCE0F1"><div align="center"><strong><font color="#002E5B"
                                    size="2" face="Verdana, Arial, Helvetica, sans-serif">ORGANIZACION</font></strong></div></td>
                                <td width="113" bgcolor="#DCE0F1"><div align="center"><strong><font color="#002E5B"
                                    size="2" face="Verdana, Arial, Helvetica, sans-serif">FECHA TRANS. </font></strong></div></td>

```

```

</tr>
<?php
    //Para ver registros
    $Contador = 1;
    $result = pg_exec(" SELECT E.codigo_egreso, E.anio_egreso, UF.descripcion_ufinal,
    O.nombre_organizacion, E.fecha_transaccion
    FROM egreso E, usuario_final UF, organizacion O
    WHERE          E.codigo_ufinal = UF.codigo_ufinal
    AND            E.codigo_organizacion = O.codigo_organizacion");
    $filas = pg_num_rows($result);
    for($i=0;$i<$filas;$i++){
        $dato1 = pg_result($result,$i,0);
        $dato2 = pg_result($result,$i,1);
        $dato3 = pg_result($result,$i,2);
        $dato4 = pg_result($result,$i,3);
        $dato5 = pg_result($result,$i,4);
        if(ceil($Contador/2) == ($Contador/2)){
            echo "<tr bgcolor=#FEF3FE' class='TextoTabla'>
            <td><div align='left'><a href='Manten_Egreso.php?CedCli=$dato1'>($dato2)-
            $dato1</a></div></td>
            <td><div align='left'>$dato3</div></td>
            <td><div align='left'>$dato4</div></td>
            <td><div align='left'>$dato5</div></td></tr>";
        }else{
            echo "<tr bgcolor='#FFFFFF' class='TextoTabla'>
            <td><div align='left'><a href='Manten_Egreso.php?CedCli=$dato1'>($dato2)-
            $dato1</a></div></td>
            <td><div align='left'>$dato3</div></td>
            <td><div align='left'>$dato4</div></td>
            <td><div align='left'>$dato5</div></td></tr>";
        }
        Contador++;
    }
    pg_FreeResult($result);

?>
</table>
</div></td></tr>
</table>
<br>
<form name="form1" method="post" action="Egresos.php">
<table width="200" border="0" align="center" cellpadding="0" cellspacing="0">
<tr>
<td><div align="center">
<input type="submit" name="Submit" value="Volver">
</div></td></tr>
</table>
</form></td></tr>
</table>
</body>
</html>

```

#### Envio\_Egresos.php

```

<html>
<head>
<title>Untitled Document</title>
</head>
<body>
<?php
    $Anio = $_POST[txt_Anio];
    $Responsable = $_POST[cmb_Responsable];
    $UFinal = $_POST[cmb_UFinal];
    $Ubicacion = $_POST[cmb_Ubicacion];
    $Organizacion = $_POST[cmb_Organizacion];
    $FechaTE = $_POST[txt_FechaT];
    $DetalleTE = $_POST[txt_DetalleT];

```



```

$CodigoF = $_POST[txt_CodigoF];
$TDocumento = $_POST[cmb_TDocumento];
$TTransaccion = $_POST[cmb_TTransaccion];
include 'Conexion.php';
$v_codigoegre = pg_exec("select nextval('seq_cod_egre')");
$filaSec = pg_numrows($v_codigoegre);
$v_codigoegre1 = pg_result($v_codigoegre,0,0);
    // Inserta los datos a la tabla
    $sql = "INSERT INTO egreso (codigo_egreso, anio_egreso, codigo_responsable,
    codigo_ufinal, codigo_ubicacion, codigo_organizacion, fecha_transaccion,
    detalle_transaccion, codigo_funcionaria, codigo_tdocumento, codigo_ttransaccion)
    VALUES ($v_codigoegre1,$Anio', $Responsable, $UFinal, $Ubicacion, $Organizacion,
    '$FechaTE', '$DetalleTE', $CodigoF, $TDocumento, $TTransaccion)";
    pg_exec($conexion, $sql);
    echo "$sql";
?>
<?php
    echo "<SCRIPT LANGUAGE='javascript'>
        location.href = 'Ingresos_ProductosE.php?Egre=$v_codigoegre1&AnioE=$Anio';
    </SCRIPT>";
?>
</body>
</html>

```

#### Ingresos\_ProductosE.php

```

<?php
    $connstr = "dbname=SICCDD user=postgres password=123
    host=localhost port=5432";
    $conexion = pg_connect($connstr); //Selecciona la base de Hotel
    $CodigoEg = $_HTTP_GET_VARS['Egre'];
    $AnioEg = $_HTTP_GET_VARS['AnioE'];
    $result = pg_exec("SELECT E.codigo_egreso, E.anio_egreso, R.codigo_responsable,
    R.nombre_responsable, R.apellido_responsable, UF.codigo_ufinal, UF.descripcion_ufinal,
    O.codigo_organizacion, O.nombre_organizacion, E.fecha_transaccion, E.detalle_transaccion,
    TD.codigo_tdocumento, TD.descripcion_tdocumento, OT.codigo_ttransaccion,
    OT.descripcion_ttransaccion, F.codigo_funcionaria
    FROM egreso E, responsable R, usuario_final UF, organizacion O, tipodocumento TD,
    objetotransaccion OT, funcionaria F
    WHERE E.codigo_responsable = R.codigo_responsable
    AND E.codigo_ufinal = UF.codigo_ufinal
    AND E.codigo_organizacion = O.codigo_organizacion
    AND E.codigo_tdocumento = TD.codigo_tdocumento
    AND E.codigo_ttransaccion = OT.codigo_ttransaccion
    AND E.codigo_funcionaria = F.codigo_funcionaria
    AND E.codigo_egreso = ".$CodigoEg." AND E.anio_egreso = ".$AnioEg."");
    $filas=pg_numrows($result);
?>
<html>
<head>
<title> </title>
</head>
<body>
<div align="right"></div>
<table width="607" border="0" align="center" cellpadding="0" cellspacing="0" bgcolor="#FFFFFF">
<tr>
<td width="607" height="453"><div align="center">
<table width="555" border="0" align="center" cellpadding="0" cellspacing="0">
<tr>
<td><form name="form2" method="post" action="Ingresos_ProductosE.php?Egre=<?php echo
"$CodigoEg"; ?>&AnioE=<?php echo "$AnioEg"; ?>">
<table width="100%" border="0" align="center" cellpadding="0" cellspacing="0">
<tr>
<td height="23" colspan="7"><div align="center"><span class="Estilo23">CABECERA
</span></div></td></tr>

```

```

<tr bgcolor="#6CCFFF">
  <td bgcolor="#FBEADB"><div align="right"><span class="Estilo23">Codigo Transaccion:
    </span></div></td>
  <td bgcolor="#FFFFFF">&nbsp;</td>
  <td bgcolor="#FFFFFF"><div align="left"><span class="Estilo25"><font color="#002E5B"
    size="2" face="Verdana, Arial, Helvetica, sans-serif">
<?php
  $CodigoE1= pg_result($result, 0, 0);
  echo "$CodigoE1";
  echo "<input name='txt_CodigoE' type='hidden' id='txt_CodigoE' value='$CodigoE1'>";
?>
</font> </span></div></td>
  <td bgcolor="#FFFFFF">&nbsp;</td>
  <td bgcolor="#FFFFFF">&nbsp;</td>
  <td bgcolor="#FFFFFF">&nbsp;</td>
  <td bgcolor="#FFFFFF">&nbsp;</td></tr>
<tr bgcolor="#6CCFFF">
  <td width="18%" bgcolor="#0091D7"><div align="right" class="Estilo23">A&ntilde;o
    Transaccion:</div></td>
  <td width="2%" bgcolor="#6CCFFF">&nbsp;</td>
  <td width="32%" bgcolor="#6CCFFF"><font color="#002E5B" size="2" face="Verdana, Arial,
    Helvetica, sans-serif">
  <span class="Estilo25"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
    serif">
  <?php
    $AnioE1= pg_result($result, 0, 1);
    echo "$AnioE1";
    echo "<input name='txt_Anio' type='hidden' id='txt_Anio' value='$AnioE1'>";
  ?>
  </font></span> </font></td>
  <td width="3%" bgcolor="#FFFFFF">&nbsp;</td>
  <td width="15%" bgcolor="#0091D7"><div align="right" class="Estilo23">Fecha:</div></td>
  <td width="2%" bgcolor="#6CCFFF">&nbsp;</td>
  <td width="28%" bgcolor="#6CCFFF"> <font color="#002E5B" size="2" face="Verdana, Arial,
    Helvetica, sans-serif">
  <span class="Estilo25"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
    serif">
  <font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">
  <?php
    $FechaTE1= pg_result($result, 0, 9);
    echo "$FechaTE1";
    echo "<input name='txt_FechaTE' type='hidden' id='txt_FechaTE' value='$FechaTE1'>";
  ?>
  </font> </font></span> </font></td></tr>
<tr>
  <td bgcolor="#FBEADB"><div align="right" class="Estilo23">Responsable:</div></td>
  <td>&nbsp;</td>
  <td><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">
  <span class="Estilo25"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
    serif">
  <?php
    $CodigoR1= pg_result($result, 0, 2);
    $NombreR1= pg_result($result, 0, 3);
    $ApellidoR1= pg_result($result, 0, 4);
    echo "$ApellidoR1 $NombreR1";
    echo "<input name='txt_Responsable' type='hidden' id='txt_Responsable'
    value='$CodigoR1'>";
  ?>
  </font></span> </font></td>
  <td bgcolor="#FFFFFF">&nbsp;</td>
  <td bgcolor="#FBEADB"><div align="right" class="Estilo23">Tipo de Documento:</div>
  <div align="right"></div></td>
  <td>&nbsp;</td>

```

```

<td class="Estilo25"><div align="left"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">
<font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">
<?php
    $CodigoTD1= pg_result($result, 0, 11);
    $NombreTD1= pg_result($result, 0, 12);
    echo "$NombreTD1";
    echo "<input name='txt_TDocumento' type='hidden' id='txt_TDocumento'
    value='$CodigoTD1'>";
?>
</font> </font></div></td></tr>
<tr bgcolor="#6CCFFF">
<td bgcolor="#0091D7"><div align="right" class="Estilo25"><strong>Organizacion:
</strong></div></td>
<td bgcolor="#6CCFFF">&nbsp;</td>
<td bgcolor="#6CCFFF"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">
<span class="Estilo25"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">
<?php
    $CodigoO1= pg_result($result, 0, 7);
    $NombreO1= pg_result($result, 0, 8);
    echo "$NombreO1";
    echo "<input name='txt_Organizacion' type='hidden' id='txt_Organizacion' value=
    '$CodigoO1'>";
?>
</font></span> </font></td>
<td bgcolor="#FFFFFF">&nbsp;</td>
<td bgcolor="#6CCFFF"><div align="right"><span class="Estilo25"><span class="Estilo23">Tipo
de Transaccion:</span></span></div></td>
<td bgcolor="#6CCFFF">&nbsp;</td>
<td bgcolor="#6CCFFF"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">
<span class="Estilo25"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">
<font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">
<?php
    $CodigoOT1= pg_result($result, 0, 13);
    $NombreOT1= pg_result($result, 0, 14);
    echo "$NombreOT1";
    echo "<input name='txt_TTransaccion' type='hidden' id='txt_TTransaccion' value='$CodigoOT1'>";
?>
</font> </font></span> </font></td>
</tr>
<tr bgcolor="#6CCFFF">
<td bgcolor="#0091D7"><div align="right"><span class="Estilo25"><strong>Usuario Final:
</strong></span></div></td>
<td bgcolor="#6CCFFF">&nbsp;</td>
<td bgcolor="#6CCFFF"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">
<span class="Estilo25"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">
<font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif"><font color="#002E5B"
size="2" face="Verdana, Arial, Helvetica, sans-serif">
<?php
    $CodigoUF1= pg_result($result, 0, 5);
    $NombreUF1= pg_result($result, 0, 6);
    echo "$NombreUF1";
    echo "<input name='txt_UFinal' type='hidden' id='txt_UFinal' value='$CodigoUF1'>";
?>
</font>
</font></font></span> </font></td>
<td bgcolor="#FFFFFF">&nbsp;</td>
<td bgcolor="#6CCFFF"><div align="right"><span class="Estilo25">
<strong>Detalle:</strong></span></div></td>
<td bgcolor="#6CCFFF">&nbsp;</td>
<td bgcolor="#6CCFFF"><div align="left"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif"><span class="Estilo25"><font color="#002E5B" size="2"

```

```

        face="Verdana, Arial, Helvetica, sans-serif"><font color="#002E5B" size="2"
        face="Verdana, Arial, Helvetica, sans-serif">
    <?php
        $DetalleTE1= pg_result($result, 0, 10);
        $CodigoF1= pg_result($result, 0, 15);
        echo "$DetalleTE1";
        echo "<input name='txt_DetalleTE' type='hidden' id='txt_DetalleTE' value='$DetalleTE1'>";
        echo "<input name='txt_CodigoF' type='hidden' id='txt_CodigoF' value='$CodigoF1'>";
    ?>
    </font></font></span></font></div></td>
</tr>
<tr>
    <td height="52" colspan="7"><div align="center">
        <hr>
    <?php
        $CodigoEg1 = $HTTP_GET_VARS['Egre'];
        $AnioEg1 = $HTTP_GET_VARS['AnioE'];
        $Producto = $_POST[cmb_Producto];
        $Cantidad = $_POST[txt_Cantidad];
        $Contador = 1;
        $v_CodigoEggr = pg_exec("select nextval('seq_cod_degre')");
        $filaSec = pg_numrows($v_CodigoEggr);
        $v_codigodegre11 = pg_result($v_CodigoEggr,0,0);
        if($Producto != ""){
            $sql ="INSERT INTO detalle_egreso (codigo_degreso, codigo_egreso, anio_egreso,
            codigo_producto, cantidad) VALUES ($v_codigodegre11, $CodigoEg1, '$AnioEg1',
            '$Producto', $Cantidad)";
            pg_exec($conexion, $sql);
        }
    ?>
<table width="549" border="0" cellpadding="0" cellspacing="0">
    <tr>
        <td height="26" colspan="7"><div align="right" class="Estilo25">
            <div align="center"><strong>PRODUCTOS</strong></div>
        </div></td>
    </tr>
    <tr class="Estilo25">
        <td width="87" class="Estilo25"><div align="right" class="Estilo25">
            <div align="right"><strong>Producto:</strong></div>
        </div></td>
        <td width="9"><div align="left">
        </div></td>
        <td width="250"><div align="left">
            <?php
                $resultP1 = pg_exec("SELECT codigo_producto, descripcion_producto
                FROM producto
                order by descripcion_producto");
                $filasP1 = pg_num_rows($resultP1);
                echo "<select name='cmb_Producto'>";
                for($i=0;$i<$filasP1;$i++)
                {
                    $dato1_1 = pg_result($resultP1,$i,0);
                    $dato2_1 = pg_result($resultP1,$i,1);
                    echo "<option value='$dato1_1'>$dato2_1</option>";
                }
                echo "</select>";
            ?>
        </div></td>
        <td width="10"><div align="right"></div></td>
        <td width="109"><div align="right"></div></td>
        <td width="7">&nbsp;</td>
        <td width="77"><div align="left"></div></td>
    </tr>
    <tr class="Estilo25">
        <td width="87"><div align="right"><strong>Descripci&ocaron;</strong></div></td>

```

```

        <td><div align="left"></div></td>
        <td><div align="left">
            <input name="txt_DescripcionProd" type="text" id="txt_DescripcionProd" size="40"
            maxlength="50">
        </div></td>
        <td><div align="right"></div></td>
        <td><div align="right"></div></td>
        <td>&nbsp;</td>
        <td><div align="left"></div></td>
    </tr>
    <tr class="Estilo25">
        <td><div align="right"><strong>Cantidad:</strong></div></td>
        <td>&nbsp;</td>
        <td><div align="left">
            <input name="txt_Cantidad" type="text" id="txt_CodigoProd3" size="10" maxlength="5">
        </div></td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
    </tr>
    <tr class="Estilo25">
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
    </tr>
    <tr class="Estilo25">
        <td colspan="7"><div align="center">
            <input name="btn_Guardar" type="submit" id="btn_Guardar" value="A&ntilde;adir
            Productos">
        </div></td></tr>
    </table>
    <hr>
    </div></td></tr>
</table>
</form></td></tr>
</table>
</div>
<form name="form1" method="post" action="Ingresos.php">
<div align="center">
<table width="563" border="1" cellpadding="0" cellspacing="0">
<tr bordercolor="#002E5B" bgcolor="#DCE0F1">
<td width="75"><div align="center"><strong><font color="#002E5B" size="2" face="Verdana, Arial,
Helvetica, sans-serif">CODIGO</font></strong></div></td>
<td width="383" bgcolor="#DCE0F1"><div align="center"><strong><font color="#002E5B" size="2"
face="Verdana, Arial, Helvetica, sans-serif">DESCRIPCION</font></strong></div></td>
<td width="97" bgcolor="#DCE0F1"><div align="center"><strong><font color="#002E5B" size="2"
face="Verdana, Arial, Helvetica, sans-serif">CANTIDAD</font></strong></div></td>
</tr>
<?php
    $Contador = 1;
    //Hace una consulta para ver si cliente existe
    $resultIn = pg_exec("SELECT E.codigo_egreso, E.anio_egreso, DE.codigo_producto,
    P.descripcion_producto, DE.cantidad
    FROM detalle_egreso DE, egreso E, producto P
    WHERE DE.codigo_egreso = E.codigo_egreso
    AND DE.anio_egreso = E.anio_egreso
    AND DE.codigo_producto = P.codigo_producto
    AND DE.codigo_egreso = $CodigoEg
    AND DE.anio_egreso = '$AnioEg'");
    $filasIn = pg_num_rows($resultIn);

```

```

        for($i=0;$i<$filasLn;$i++){
            $datol1 = pg_result($resultLn,$i,2);
            $datol2 = pg_result($resultLn,$i,3);
            $datol3 = pg_result($resultLn,$i,4);
            if(ceil($Contador/2) == ($Contador/2)){
                echo "<tr bgcolor='#FEF3FE' class='TextoTabla'>
                <td><div align='left'><a href='Manten_DEgreso.php?CedCli=$datol1'>$datol1
                </a></div></td>
                <td><div align='left'>$datol2</div></td>
                <td><div align='left'>$datol3</div></td></tr>";
            }else{
                echo "<tr bgcolor='#FFFFFF' class='TextoTabla'>
                <td><div align='left'><a href='Manten_DEgreso.php?CedCli=$datol1'>$datol1
                </a></div></td>
                <td><div align='left'>$datol2</div></td>
                <td><div align='left'>$datol3</div></td></tr>";
            }
            $Contador++;
        }
        pg_FreeResult($result);

    ?>
</table>
<br>
<input type="submit" name="Submit" value="Finalizar">
</div>
</form>
<p align="center">&nbsp;</p></td> </tr>
</table>
</body>
</html>

```

#### Evaluacion.php

```

<?php
session_start();
if (isset($_SESSION['k_username'])) {
    $usuario1 = $_SESSION['k_username'];
    //echo "zdiego.$usuario1";
}else{
    echo "<TABLE cellSpacing=0 cellPadding=0 width=718 border=0 align=center>
    <TBODY>
    <TR>
    <TD><IMG height=62
    src='Graficos/logo.gif' width=177></TD>
    <TD width=541 bgColor=#9a4d9f><div align='center'><font face = 'Verdana' color = 'white' size = '3'>
    <strong>DEBE INICIAR LA SESION CON UNA CUENTA EXISTENTE</strong>
    </font></div></TD>
    </TR>
    </TABLE>
    <div align='center'><font face = 'Verdana' color = 'white' size = '2'><strong><a href='Index.html'>
    REGRESAR A PRINCIPAL</a></strong></font></div>";
    exit();
}
?>
<html>
<head>
<title>Evaluacion CENDOC - SISCON</title>
</head>
<body>
<?php
include 'Conexion.php';
$Fecha1 = getdate();
$FechaE = $Fecha1[year]."-".$Fecha1[mon]."-".$Fecha1[mday];
?>
<TABLE width=592 height=284 border=0 align=center cellPadding=0 cellSpacing=0>

```

```

<TBODY>
<TR>
<TD vAlign=top width=501><TABLE height=64 cellSpacing=0 cellPadding=0 width=576 align=center
border=0>
<TBODY>
<TR>
<TD width=576 height=64><form action="Ingreso_Evaluacion.php?valor=1" method="post"
name="frm_UsuarioFinal" id="frm_UsuarioFinal">
<p align="center"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
serif"><strong>EVALUACION CENTRO DE DOCUMENTACION</strong></font></p>
<div align="right">
<table width="585" border="0" cellpadding="0" cellspacing="0">
<tr>
<td width="175"><div align="right"><font color="#002E5B" size="2" face="Verdana, Arial,
Helvetica, sans-serif">CEDULA:</font></div></td>
<td width="401">
<div align="left">
<input name="txt_CedulaU" type="text" id="txt_CedulaU" size="15" maxlength="10">
</div></td></tr>
<tr>
<td><p align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
serif">NOMBRES:</font></p></td>
<td>
<div align="left">
<input name="txt_NombresU" type="text" id="txt_NombresU" size="27" maxlength="25">
</div></td></tr>
<tr>
<td><div align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
serif">APELLIDOS:</font></div></td>
<td>
<div align="left">
<input name="txt_ApellidosU" type="text" id="txt_ApellidosU" size="27" maxlength="25">
</div></td></tr>
<tr>
<td><div align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
serif">SEXO:</font></div></td>
<td>
<div align="left">
<select name="cmb_Sexo" id="cmb_Sexo">
<option value="Femenino">Femenino</option>
<option value="Masculino">Masculino</option>
</select>
</div></td></tr>
<tr>
<td><div align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
serif">INSTITUCION:</font></div></td>
<td>
<div align="left">
<input name="txt_InstitucionU" type="text" id="txt_InstitucionU" size="60"
maxlength="70">
</div></td></tr>
<tr>
<td><div align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
serif">EMAIL:</font></div></td>
<td>
<div align="left">
<input name="txt_EmailU" type="text" id="txt_EmailU" size="60" maxlength="50">
</div></td></tr>
<tr>
<td><div align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
serif">TEMA DE CONSULTA:</font></div></td>
<td>
<div align="left">

```

```

        <textarea name="txt_TemaCon" cols="50" rows="4" id="txt_TemaCon"></textarea>
    </div></td></tr>
<tr>
<td height="28"><div align="right"><font color="#002E5B" size="2" face="Verdana, Arial,
Helvetica, sans-serif">TELEFONO:</font></div></td>
<td>
<div align="left">
<input name="txt_TelefonoU" type="text" id="txt_TelefonoU" size="22" maxlength="20">
</div></td></tr>
<tr>
<td><div align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
serif">FECHA CONSULTA:</font></div></td>
<td><div align="left"><strong><font color="#002E5B" size="2" face="Verdana, Arial,
Helvetica, sans-serif"><?php echo "<input name='txt_FechaCon' type='hidden'
id='txt_FechaCon' value='$FechaE'>$FechaE"; ?></font>
</strong></div></td></tr>
<tr>
<td><div align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
serif">TIPO DE USUARIO:</font></div></td>
<td><div align="left">
<?php
    $result = pg_exec("SELECT codigo_tusuario, descripcion_tusuario
FROM tipo_usuario
order by codigo_tusuario");
    $filas = pg_num_rows($result);
    echo "<select name='cmb_TUusuario'>";
    for($i=0;$i<$filas;$i++)
    {
        $dato1 = pg_result($result,$i,0);
        $dato2 = pg_result($result,$i,1);
        echo "<option value='$dato1'>$dato2</option>";
    }
    echo "</select>";
    ?>
</div></td></tr>
<tr>
<td><div align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-
serif">TIPO DE SERVICIO:</font></div></td>
<td><div align="left">
<?php
    $result1 = pg_exec("SELECT codigo_tservicio, descripcion_tservicio
FROM tipo_servicio
order by codigo_tservicio");
    $filas1 = pg_num_rows($result1);
    echo "<select name='cmb_TServicio'>";
    for($i=0;$i<$filas1;$i++)
    {
        $dato1_1 = pg_result($result1,$i,0);
        $dato2_1 = pg_result($result1,$i,1);
        echo "<option value='$dato1_1'>$dato2_1</option>";
    }
    echo "</select>";
    ?>
</div></td></tr>
</table>
</div>
<table width="260" border="0" align="center" cellpadding="0" cellspacing="0">
<tr>
<td width="260"><strong><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica,
sans-serif"><?php echo "<input name='txt_CodigoF' type='hidden' id='txt_CodigoF'
value='$usuario1'>"; ?></font></strong></td>
<td width="260">&nbsp;</td></tr>
<tr>
<td><div align="center">

```





```

$CodigoF = $_POST[txt_CodigoF];
$v_codigouser = pg_exec("select nextval('seq_cod_eval')");
$filaSec = pg_numrows($v_codigouser);
$v_codigouser1 = pg_result($v_codigouser,0,0);
$Contador = 1;
//Hace una consulta para ver si cliente existe
$result=pg_exec("SELECT * FROM registro_user where codigo_user = $v_codigouser1");
//Ve el número de filas que obtiene el select si es 0 el cliente no existe
$filas=pg_numrows($result);
if($filas == 0){
//echo " cliente no existe";
// Inserta los datos a la tabla
sql = "INSERT INTO registro_user (codigo_user, cedula_user, nombres_user, apellidos_user,
sexo_user, institucion_user, email_user, tema_consulta, telefono_user, fecha_consulta,
codigo_tusuario, codigo_tservicio, codigo_funcionaria)
VALUES ($v_codigouser1,$CedulaU,$NombresU,
'$ApellidosU','$Sexo','$InstitucionU','$EmailU','$TemaCon','$TelefonoU','$FechaCon', $TUsuario,
$TServicio, $CodigoF)";
pg_exec($conexion, $sql);
// Muestra los datos de la tabla luego del ingreso del registro
$result = pg_exec("SELECT codigo_user, nombres_user, apellidos_user, fecha_consulta FROM
registro_user order by codigo_user");
$filas = pg_num_rows($result);
for($i=0;$i<$filas;$i++){
$dato1 = pg_result($result,$i,0);
$dato2 = pg_result($result,$i,1);
$dato3 = pg_result($result,$i,2);
$dato4 = pg_result($result,$i,3);
if(ceil($Contador/2) == ($Contador/2)){
echo "<tr bgcolor=#FEF3FE' class='TextoTabla'>
<td width='78' ><div align='left'><a href='Manten_Evaluacion.php?CedCli=$dato1'>$dato1
</a></div></td>
<td width='355'><div align='left'>$dato2 $dato3</div></td>
<td width='106'><div align='left'>$dato4</div></td></tr>";
}else{
echo "<tr bgcolor='#FFFFFF' class='TextoTabla'>
<td width='78' ><div align='left'><a href='Manten_Evaluacion.php?CedCli=$dato1'>$dato1
</a></div></td>
<td width='355'><div align='left'>$dato2 $dato3</div></td>
<td width='106'><div align='left'>$dato4</div></td></tr>";
}$Contador++;
}
}else{
echo "<p align='center' class='TextoTabla1'>Funcionaria ya existe: Datos no se grabaron</p>";
}}
//Para Modificar registros
if($valor1==2)
{
$CedulaU = $_POST[txt_CedulaU];
$NombresU = $_POST[txt_NombresU];
$ApellidosU = $_POST[txt_ApellidosU];
$Sexo = $_POST[cmb_Sexo];
$InstitucionU = $_POST[txt_InstitucionU];
$EmailU = $_POST[txt_EmailU];
$TemaCon = $_POST[txt_TemaCon];
$TelefonoU = $_POST[txt_TelefonoU];
$FechaCon = $_POST[txt_Fechacon];
$TUsuario = $_POST[cmb_TUsuario];
$TServicio = $_POST[cmb_TServicio];
$CodigoF = $_POST[txt_CodigoF];
$Contador = 1;
//Modifica los datos a la tabla

```

```

$sql = "UPDATE funcionaria SET nombre_funcionaria = '$NombresF', apellido_funcionaria =
'$ApellidosF', direccion = '$DireccionF', telefono = '$TelefonoF', celular = '$CelularF' WHERE
cedula_funcionaria = '$CedulaF'";
pg_exec($conexion, $sql);
// Muestra los datos de la tabla luego del ingreso del registro
$result = pg_exec("SELECT codigo_user, nombres_user, apellidos_user, fecha_consulta FROM
registro_user order by codigo_user");
$filas = pg_num_rows($result);
for($i=0;$i<$filas;$i++){
    $dato1 = pg_result($result,$i,0);
    $dato2 = pg_result($result,$i,1);
    $dato3 = pg_result($result,$i,2);
    $dato4 = pg_result($result,$i,3);
    if(ceil($Contador/2) == ($Contador/2)){
        echo "<tr bgcolor='#FEF3FE' class='TextoTabla'>
        <td width='78' ><div align='left'><a href='Manten_Evaluacion.php?CedCli=$dato1'>$dato1
        </a></div></td>
        <td width='355'><div align='left'>$dato2 $dato3</div></td>
        <td width='106'><div align='left'>$dato4</div></td></tr>";
    }else{
        echo "<tr bgcolor='#FFFFFF' class='TextoTabla'>
        <td width='78' ><div align='left'><a href='Manten_Evaluacion.php?CedCli=$dato1'>$dato1
        </a></div></td>
        <td width='355'><div align='left'>$dato2 $dato3</div></td>
        <td width='106'><div align='left'>$dato4</div></td></tr>";
    }
    $Contador++;
}
}
Para ver registros
if($valor1==3)
{
    $CedulaU = $_POST[txt_CedulaU];
    $NombresU = $_POST[txt_NombresU];
    $ApellidosU = $_POST[txt_ApellidosU];
    $Sexo = $_POST[cmb_Sexo];
    $InstitucionU = $_POST[txt_InstitucionU];
    $EmailU = $_POST[txt_EmailU];
    $TemaCon = $_POST[txt_TemaCon];
    $TelefonoU = $_POST[txt_TelefonoU];
    $FechaCon = $_POST[txt_Fechacon];
    $TUsuario = $_POST[cmb_TUsuario];
    $TServicio = $_POST[cmb_TServicio];
    $CodigoF = $_POST[txt_CodigoF];
    $Contador = 1;
    // Muestra los datos de la tabla luego del ingreso del registro
    $result = pg_exec("SELECT codigo_user, nombres_user, apellidos_user, fecha_consulta FROM
    registro_user order by codigo_user");
    $filas = pg_num_rows($result);
    for($i=0;$i<$filas;$i++){
        $dato1 = pg_result($result,$i,0);
        $dato2 = pg_result($result,$i,1);
        $dato3 = pg_result($result,$i,2);
        $dato4 = pg_result($result,$i,3);
        if(ceil($Contador/2) == ($Contador/2)){
            echo "<tr bgcolor='#FEF3FE' class='TextoTabla'>
            <td width='78' ><div align='left'><a href='Manten_Evaluacion.php?CedCli=$dato1'>$dato1
            </a></div></td>
            <td width='355'><div align='left'>$dato2 $dato3</div></td>
            <td width='106'><div align='left'>$dato4</div></td></tr>";
        }else{
            echo "<tr bgcolor='#FFFFFF' class='TextoTabla'>
            <td width='78' ><div align='left'><a href='Manten_Evaluacion.php?CedCli=$dato1'>$dato1
            </a></div></td>
            <td width='355'><div align='left'>$dato2 $dato3</div></td>

```



```

<td><p align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">NOMBRES:</font></p></td>
<td> <div align="left"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">
  <?php $NombresU1= pg_result($result, 0, 2);
      echo "$NombresU1";
  ?>
</font> </div></td></tr>
<tr>
<td><div align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">APELLIDOS:</font></div></td>
<td><div align="left"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">
  <?php $ApellidosU1= pg_result($result, 0, 3);
      echo "$ApellidosU1";
  ?>
</font> </div></td></tr>
<tr>
<td><div align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">SEXO:</font></div></td>
<td> <div align="left"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">
  <?php $SexoU1= pg_result($result, 0, 4);
      echo "$SexoU1";
  ?>
</font> </div></td></tr>
<tr>
<td><div align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">INSTITUCION:</font></div></td>
<td> <div align="left"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">
  <?php $InstitucionU1= pg_result($result, 0, 5);
      echo "$InstitucionU1";
  ?>
</font> </div></td></tr>
<tr>
<td><div align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">EMAIL:</font></div></td>
<td> <div align="left"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">
  <?php $EmailU1= pg_result($result, 0, 6);
      echo "$EmailU1";
  ?>
</font> </div></td></tr>
<tr>
<td><div align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">TEMA DE CONSULTA:</font></div></td>
<td><div align="left"><span class="Estilo25"><font color="#002E5B" size="2"
  face="Verdana,Arial, Helvetica, sans-serif">
  <?php
    $TemaU1= pg_result($result, 0, 7);
    echo "<textarea name='txt_TemaCon' cols='50' rows='4' id='textarea'>$TemaU1
    </textarea>";
  ?>
  </font></span></div></td></tr>
<tr>
<td><div align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">TELEFONO:</font></div></td>
<td><div align="left"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">
  <?php $TelefonoU1= pg_result($result, 0, 8);
      echo "$TelefonoU1";
  ?>
</font></div></td></tr>

```

```

<tr>
<td><div align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">FECHA DE CONSULTA:</font></div></td>
<td><div align="left"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">
<?php $FechaU1= pg_result($result, 0, 12);
echo "$FechaU1";
?>
</font></div></td></tr>
<tr>
<td><div align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">TIPO DE USUARIO:</font></div></td>
<td><div align="left"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">
<?php $TUsuarioU1= pg_result($result, 0, 9);
$result2 = pg_exec("SELECT descripcion_tusuario FROM tipo_usuario where
codigo_tusuario = $TUsuarioU1");
$filas2 = pg_num_rows($result2);
$dato2_1 = pg_result($result2,$i,0);
echo "$dato2_1";
?>
</font></div></td>
</tr>
<tr>
<td><div align="right"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">TIPO DE SERVICIO:</font></div></td>
<td><div align="left"><font color="#002E5B" size="2" face="Verdana, Arial, Helvetica, sans-serif">
<?php $TServicioU1= pg_result($result, 0, 10);
$result3 = pg_exec("SELECT descripcion_tservicio FROM tipo_servicio where
codigo_tservicio = $TServicioU1");
$filas3 = pg_num_rows($result3);
$dato3_1 = pg_result($result3,$i,0);
echo "$dato3_1";
?>
</font></div></td></tr>
</table>
<?php
pg_FreeResult($result);
?>
<table width="260" border="0" cellpadding="0" cellspacing="0">
<tr>
<td width="260">&nbsp;</td>
</tr>
<tr>
<td><div align="center">
<input name="btn_Grabar" type="submit" id="btn_Grabar" value="Ver Evaluaciones">
</div></td>
</tr>
</table>
</div>
</form>
</TD>
</TR> </TBODY> </TABLE>
<DIV align=center></DIV></TD>
</TR> </TBODY></TABLE>
</body>
</html>

```